A Cost-optimal Natural Gas Contract Selection for the Day-Ahead Planning

A Thesis submitted in partial fulfillment of requirements
for the Degree of Master of Science

Computer Science and Media

Submitted by:
Mario Wandpflug
Mat.-Nr. 29128

Supervisor	Prof. Dr. Roland Schmitz (HdM Stutgart)
First Co-Advisor	M. Sc. Michel Zedler (EXXETA AG)
Second Co-Advisor	M. Sc. Andreas Bräsigk (EXXETA GmbH)

November 9th, 2016

Declaration of Originality

Hereby I solemnly declare, that

- this Master Thesis is entirely the product of my own scholarly work, unless otherwise indicated in the text or references, or acknowledged below
- I have indicated the thoughts adopted directly or indirectly from other sources at the appropriate places within the document
- this Master Thesis has not been submitted either in whole or part, for a degree at this or any other university or institution
- I have not published this Master Thesis in the past
- the printed version is equivalent to the submitted electronic one
- I am aware that a dishonest declaration will entail legal consequences

Stuttgart, November 9th, 2016 ______________________________

Mario Wandpflug

Abstract

Before gas is transported, natural gas traders have to plan with many contracts every day. If a cost-optimized solution is sought the most attractive contracts of a large contract set have to be selected. This kind of cost-optimization is also known as day-ahead balancing problem. In this work it is shown that it is possible to express this problem as a linear program that considers important influences and restrictions in the daily trading.
The aspects of the day-ahead balancing problem are examined and modelled individually. This way a basic linear program is gradually adapted towards a realistic mathematical formulation. The resulting linear optimization problem is implemented as a prototype that considers the discussed aspects of a cost-optimized contract selection.

Keywords: Operations Research, Linear Programming, Mixed-Integer Linear Programming, Natural Gas Trading, Cost-optimal contract selection

Kurzfassung

Um Lieferzusagen zu erfüllen, planen Erdgashändler vor jedem Transporttag mit vielen Gashandelsverträgen. Falls eine kostenoptimierte Lösung erwünscht ist, müssen aus einer großen Vertragsmenge die günstigsten Optionen ausgewählt werden. Diese Art der Kostenoptimierung wird auch als Day-Ahead-Balancing Problem bezeichnet. In dieser Arbeit wird gezeigt, dass es möglich ist diesen Problemtyp in ein lineares Optimierungsproblem auszudrücken, das wichtige Einflüsse und Einschränkungen im täglichen Gashandel beachtet.
Wichtige Aspekte des Day-Ahead Balancing Problems werden einzeln betrachtet und modelliert. Auf diese Weise wird ein grundlegendes lineares Optimierungsproblem zu einer realitätsnahen mathematischen Formulierung angepasst. Resultierend aus der Modellierung wird ein Prototyp implementiert, der die diskutierten Aspekte einer kostenoptimierten Vertragsbeschäftigung berücksichtigt.

Schlagworte: Operations Research, Linear Optimierung, Gemischte ganzzahlige Optimierung, Erdgashandel, kostenoptimierte Vertragsbeschäftigung

Contents

List of Figures

List of Tables

List of Abbreviations

GLPK	GNU Linear Programming Kit
LP	linear programming
MIP	mixed-integer linear programming
OR	Operations Research
SSO	storage system operator
TSO	transmission system operator
VTP	virtual trading point

Chapter 1

Introduction

Natural gas traders are confronted with the problem of fulfilling delivery commitments to customers every day. These agreements state demand or supply amounts, which are summarized to a balancing amount. In order to satisfy this balancing amount, traders operate with concluded contracts or gas market options.
Based on a defined strategy, a trader needs to decide how much natural gas is picked of each contract for the next day. This problem can be denoted as day-ahead balancing problem. An example strategy is the minimization of costs, which means that the trader needs to choose the most attractive contracts.
A cost-minimal contract selection depends on different influencing factors, for example the contract type, pricing, contractual and strategic bounds or physical capacities in transport. All these factors need to be considered and generate a decision task with a large amount of data. Scanning large amounts of data can be hard and this task has to be executed every day.
Assuming the data to plan is known for the next day and finding a cost-minimal distribution is the primary goal, linear programming can be one approach to handle this allocation problem. The underlying linear function model forms the basis of the optimization problem and by applying the contractual input data to the linear programming problem, the function is parametrised and solved. In this way the yielded solution outputs the minimal costs and an optimal contract selection.
There are other works, that covered the optimization of physical transport of the gas network, e.g. [Mid07, TRFM07]. Other works observed certain stakeholders, such as gas storages, local distribution facilities or producers [Hol08, GW97]. Furthermore there have been general approaches for a portfolio selection [Mar52].

Since there is a need for an abstracted mathematical formulation of natural gas trading, the motivation arises to generate a linear program to solve the natural gas traders day-ahead balancing problem. This work comprises an analysis of issues and restrictions in the day-ahead planning and the formulation of a linear mathematical model. By using linear and mixed-integer programming, an optimal contract setting shall be found which yields a cost-minimal selection of contracts. This solution proposes a strategy for the trader in order to satisfy the commitments and save costs.
Chapter 2 examines important issues and discusses the day-ahead balancing problem. Changes of the natural gas market are explained and planning problem is delimited. In addition, important market participants and restrictions are discussed. The problem domain serves as foundation of the linear optimization problem. Chapter 3 proposes linear programming and mixed-integer programming as a tool to find the cost-minimal contract distribution. The first part covers the backgrounds and principles of linear programming and mixed-integer programming. The theoretical background is applied in the second part of chapter 3, where a first basic linear program exemplifies the approach to find a cost-optimal contract selection.
Since the basic linear programming problem would not be applicable for a real scenario, chapter 4 presents adaptations of this basic application, such that the main requirements of the problem description paper are regarded. An isolated view on each problem shall clarify and discuss the features of each problem. This shall help to comprehend the specific feature and the adaptation of the linear mathematical model.
Based upon the proposed mixed-integer programming problem, a prototype implementation is suggested in chapter 5. After showing briefly the design of the prototype, the experimental data structure is explained and passed to the solver in order to find a cost-minimal contract selection. Furthermore the performance is measured by passing bigger data sets to the prototype implementation.
Chapter 6 discusses the results, advantages, disadvantages and problems of the proposed mixed-integer programming problem. The last chapter 7 sums up the work and discusses further extension and future works.

Chapter 2

The Day-Ahead Balancing Problem of a Natural Gas Trader

Trading gas is a challenging process because the gas market is subjected to legal, economic and physical regulations. The gas trader's primary goal is to generate high profits revenues and keep the costs low while following all legal and market regulations. A means to reach his goal is the development of a strategy consisting of different contracts with other interest groups at the gas market. One such strategy can be the minimization of costs.

His strategy is based upon a flexible portfolio, i.e. there is a pool of concluded long-term and mid-term contracts which constitute delivery demands or supply sources. Compared to the last ten years, the gas market has become liberalized and more dynamic. Trading often occurs over the counter between traders and deals at the exchange lead to shorter planning phases. For each contract the trader has to find the optimal operational amount in order to minimize the costs.

All demand amounts for the next day need to be net out by supply amounts and vice versa. Otherwise cost-intensive imbalances could occur and a trader needs to find further supply sources, develop a day-ahead and intra-day planning with the given portfolio and short-dated contracts. Hence, traders need to plan with both, the portfolio and short-termed contracts in order to satisfy delivery demands for the daily planning.

These delivery demands will generate revenues if the natural gas trader can satisfy the contracts by purchasing gas from a set of suppliers. Different kinds of costs, such as purchasing and production costs, capacity bookings or violations of contracts and regulations have to be regarded by the trader. This leads to the problem of finding an optimal selection of supply contracts while regarding all delivery and supply constraints with the main objective to minimize costs.

2.1 Background

During the last ten years the gas market has been liberalized. Commodity and transmission of gas have been simplified, for example by establishing virtual trading points, a non-physical location for exchanging gas products [Com13, BD16, ZS09]. One advantage of a virtual trading point is that traders do not need to know about the exact gas network topology. For instance, they can disregard the whole transmission path, or they need not to know about capacities of single grid points inside the transmission network. Furthermore a virtual trading point offers a location for flexible gas products, such as physical short term contracts or gas market options [Gas16, Com13, ZS09], which complement the portfolio of the trader.
Along with a market liberalization the way to trade gas has affected the trading and dispatching process of a trader. The long lasting portfolio based planning to minimize costs and satisfy the delivery demands is not sufficient anymore. Traders try to find attractive short-dated deals and combine them with their portfolio. Attractive contracts mean that a trader tries to select a very reasonable mixture of contracts so that the costs are as low as possible while his revenues maximize.

2.2 Parameters and Planning Phases

Different gas products, either long-term, mid-term or short-term contracts, are the basis for the planning phases of a trader. Most contracts contain general parameters which can be extracted by the trader [ZS09, Cha16, BD16].

- Pricing models of natural gas contracts are the primary factor to determine revenues or costs in the planning. Different pricing models have been established in the natural gas market, which are decoupled of the natural gas amounts. For example, a trader can conclude a fixed price or an oil-indexed price with another trader. Subsection 2.6 covers a more detailed explanation of the pricing models.

- A required delivery or purchase amount of gas [Cha16]. Most times the contracts define a fixed or variable margin based upon a defined amount or estimated data [BD16]. For the latter historic data over a certain time span is used, e.g. the consumption of gas in the last two years. The amount either determines the revenue in case of delivery or costs in case of purchasing and affects the selection of contracts in the day-ahead planning, as discussed in subsection 2.5 and subsection 2.6.
- A location of operation where the gas will be purchased or delivered [Cha16, BD16]. These locations can be seen as grid points in the gas network. As described later in subsection 2.5, these grid points can have a physical location or exist virtually. Physical grid point capacities, for example transfer capacities or withdrawal capacities, need to be booked at the place of transaction which influences the costs for the trader. Note that virtual points do not have any capacities.
- Lead times can be seen as conditions, too. Lead time defines the time between initiation and execution of a process and has an influence to the balancing of a network.

These parameters need to be regarded in all planning phases. Figure 2.1 points out the portfolio, day-ahead/intra-day and dispatching planning phase. Long-term and mid-term estimations and selections are aligned in the portfolio planning phase including risks and price estimations. Based on the decisions of the previous phase the day-ahead/intra-day planning phase begins. Here the trader comprises the portfolio and gathers all registered delivery demands of the day. By selecting the right supply contracts or purchase options a trader tries to satisfy the demands, prevent imbalances in the network and follow his objective, namely reducing the costs [BD16]. Both, portfolio planning and short-term decisions, influence the day-ahead/intra-day planning phase. The dispatching phase executes all decisions of the previous phase. If imbalances occur in the dispatching phase, the daily planning has to be repeated.

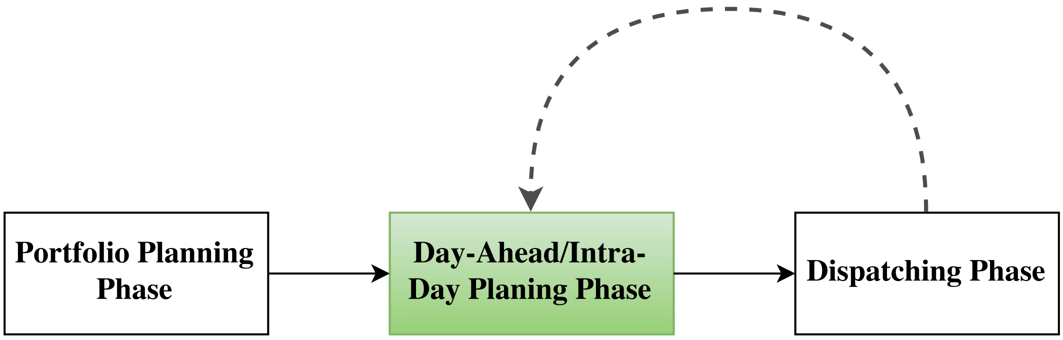

Figure 2.1: Different planning processes for a trader [BD16].

The selected contracts form the foundation and conditions for the dispatching phase, because an optimal strategy has been chosen and the transport has to be planned. During this phase the trader acquires capacities and nominates the selected contracts. Although a trader has tried to avoid imbalances in the previous phase, the balancing is done in this dispatching phase.
If any imbalances occur, the daily planning has to be reworked and executed again. The nomination is an information exchange and denotes the request of a certain gas amount at a location in the network by a trader [Ene16]. For a physical or virtual trading point the trader communicates gas quantities to transfer over a time period to the transmission system operator or market area manager respectively. Nomination contains the source and the target of transport and via the nomination an equal information level between traders is established.
During all planning phases, a natural gas trader is in touch with different interest groups. In the following subsection these interest groups are introduced and the specific roles of the trader towards them are described.

2.3 Stakeholders

Several interest groups act on the gas market. Like the trader, all groups process different jobs and follow their own interests. The stakeholders can be subdivided into the groups **commodity** and **transmission**, respectively. Figure 2.2 shows the stakeholder groups and the members of both groups. Assume that the connection between a trader and the specific stakeholder represents a contract. Concluded contracts with members of the commodity group prepare the physical movement of natural gas. For example, a trader can conclude contracts to deliver gas to a consumer or other trader, or inject gas into a storage at the next day.

The group transmission is responsible for the gas grid and thus for the physical dispatching in high-pressure gas network. The trader concludes primarily contracts concerning the physical transport of natural gas [BD16].

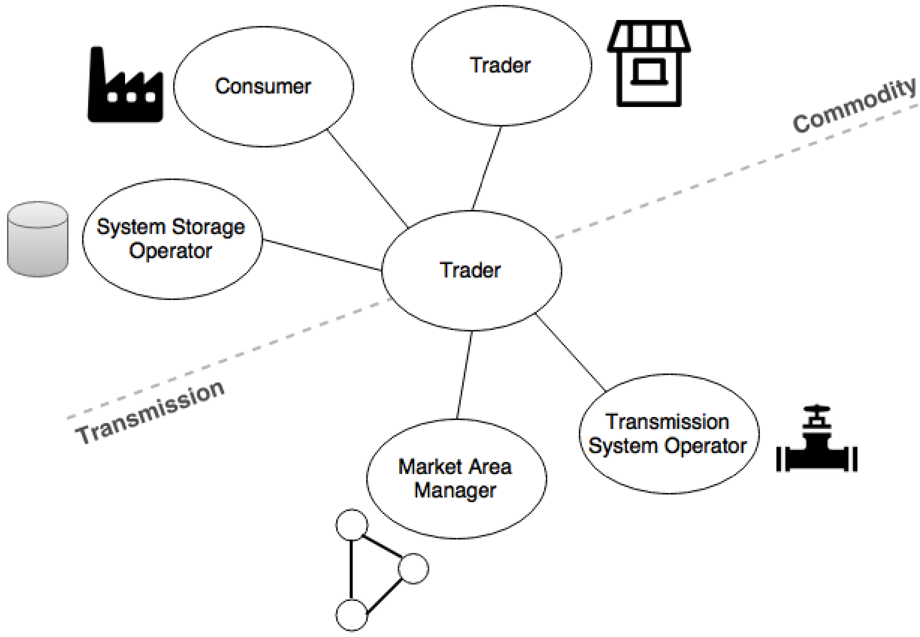

Figure 2.2: The different stakeholders of a trader divided into **commodity** and **transmission** groups. Former group contains trading partners while the latter group concentrates on dispatching [BD16].

National gas networks are subdivided into supra-regional network. This infrastructure is operated, maintained and developed by a **transmission system operator** (TSO). A TSO manages physical execution points for the transmission. He offers *entry- and exit-capacities* at these physical grid points which have to be booked by the trader if he wants to nominate the transmission at a physical points [ZS09].
A **market area manager** administers a virtual merge of several supra-regional transmission networks, called market area [Gas16]. The physical infrastructure is abstracted to network grid points such that the trading of gas is simplified between trading partners. Further remarks about the market area can be found in subsection 2.4. The market area manager establishes a trading zone with trading points, balances the gas network in the market area and publishes information for traders. To enter the market area a trader concludes a *balancing group contract* with market area manager [Gas16, BD16]. This conclusion allows the trader to use the virtual trading point as a contract mechanism and provide gas for customers. The trader becomes a *balancing group manager* from the market area manager's point of view. The trader's job is to net out gas purchases and sales of his balancing group. Differences

in sales and purchases result in imbalances which are invoiced by the market area manager. In case all traders net out imbalances in their balancing group the whole gas network will be balanced and the supplement is warranted [ZS09, BD16]. Both, the TSO and the market area manager are transmission stakeholders. The contracts between them and the trader respect mainly regulatory or physical aspects.

Besides, the transmission stakeholders the trader concludes contracts with consumers, other traders and storage system operators where the natural gas is seen as commodity. These stakeholders belong to the commodity group. The interests between them and the trader concentrate on the trading of gas.

Consumers can be distinguished into *large-scale consumers* and *small-scale consumers*. Power plants, muncipal works and industrial customers can be classified to the group of large-scale consumers. These consumers are characterized by a high consumption of gas amounts. Small-scale consumers comprise households and commercial customers. Compared to large-scale consumers they obtain lower consumption data which is reported to large-scale consumers. This means that these large-scale consumers act as an intermediary and demands natural gas of a trader. Consumers are excluded from the nomination. The trader is obliged to provide the required gas amount based on the historic measurement. In addition, a trader needs to book exit capacities with a transmission system operator because gas is withdrew from the network in the physical transmission process.

Another member of the commodity group is the **storage system operator** (SSO) which manages gas storages, e.g. cavern storages [BD16]. The conclusion of *storage contracts* allows the trader to inject into or withdraw gas out of a storage. These contracts guarantee a flexible seasonal or short-dated injection and withdrawal. There are different pricing models, which mainly affect the operational costs for withdrawal or injection of natural gas. A trader can decide whether he wants to regard stored gas in his daily planning. When gas is injected or withdrew, the trader needs to book capacities with the transmission system operator.

Last but not least other **traders** form a further interest group. There are delivery or supply contracts between two traders and there is a wide range of conditions in these contracts, e.g. by own pricing models. The motivation of trading gas depends on the trader himself. Gas producers, for instance, manage an own gas selling department and sell directly out of the gas refinery. Other traders act solely on the retail side and generate profit by buying and selling gas [BD16]. Besides, a trader could hedge against financial loss by observing the market and estimates his profits for a future time span.

All stakeholders are part of a market area and denote single grid points in the gas network. For the daily planning phases a trader has to know what types of contracts can occur at each grid point. Thus the following subsection 2.4 depicts the abstract view and explains the single grid points.

2.4 Market Area

The different gas grids form a complex transport system consisting of pipelines and different kinds of physical grid points, e.g. interconnection points between transmission system operators or withdraw points to consumers. A trader would need to know the whole path of transmission in order to satisfy the delivery demands. To simplify gas trades, market areas have been introduced [Gas16, BD16].
A market area is a virtual trading area and abstracts the commodity of gas from the physical transmission. By entering the market area, a trader needs to know the location of operation where gas is either injected (entry) or withdrew (exit) [BD16, Gas16]. Figure 2.3 illustrates the important grid points from the trader's perspective.

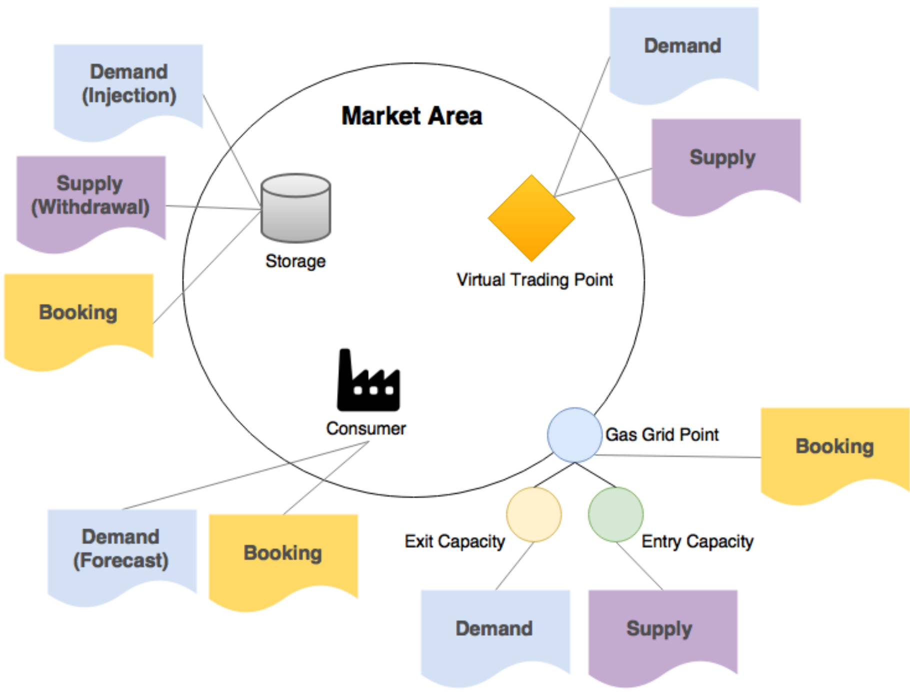

Figure 2.3: The market area denotes grid points which reveal delivery or supply contracts [BD16]. At physical grid points the trader needs to book capacities (yellow) at the TSO.

Grid points can have a physical association, such as the border grid point, storage and consumer, or no physical association like the virtual trading point. Physical associated points are entry- or exit-points. Traders usually import natural gas to the market area or export it out of the market area. These grid points can be a location for a demand or supply. The shown gas grid point in figure 2.3 can also be a location for a demand or supply as shown in 2.5.2. A TSO offers available capacities which have to be booked and nominated for dispatching [BD16].
The virtual trading point is a contractual mechanism to simplify the trading [ZS09, Gas16, BD16]. This point can be defined as operation point for transmission and it is not necessary to book capacities at the virtual trading point [Gas16]. Transmission nominations over the virtual trading point must be communicated to the responsible market area manager [Com13, Gas16, BD16].
Note that all shown grid points are operation points. These are involved in delivery or supply contracts. The delivery contracts (light blue) count to a demand and the trader withdraws gas out of the market area. Supply contracts (violet) provide gas and inject gas in the market area.
The concluded contracts obtain if the grid point is a point of delivery, supply or both. Remember that a trader desires to balance the network already in the day-ahead planning. This so called commodity balancing is important because it sets the plan of execution in the dispatching phase. In the next section a schema to balance the network is shown.

2.5 Commodity Balancing

Balancing delivery demands and supplies is an important aspect, since imbalances and violations of contract conditions generate costs. Hence a trader opposes delivery and the purchase amounts to balance them. This opposing is called **commodity balancing** [BD16]. Commodity balancing concentrates on the gas amounts. Physical capacities and constraints are already regarded so that the transport of natural gas is ensured for the next day.
Figure 2.4 shows a schema for commodity balancing. The trader extracts K known delivery and supply amounts are extracted from concluded contracts and catalogues them on the left side called *source*. All sources can lead to imbalances and the trader wants to avoid an imbalanced balancing group.

Delivery and supply contracts where the trader has to decide for the operation amounts are U unknown decision variables. These are listed on the right side of the commodity balance, called *target*. Assume that the decision variables are the most attractive contracts, i.e. less cost-intensive contracts. Via commodity balancing imbalances are revealed and the trader can estimate possible costs.
The commodity balance K^{Σ} is denoted as the sum of all known sources plus the sum of all unknown targets, as described in equation 2.1. Let k_i denote a concrete source and u_j a concrete target. Gas that flows out of the balancing group, e.g. natural gas leaves the market area via an exit point or is injected into a storage, receives a negative sign. On the contrary natural gas that flows into the balancing group receives a positive sign, for instance if natural gas enters the balancing group via an entry point or is withdrew out of a storage. This means that any source k and any target u can either be positive or negative. A balanced network will be accomplished if K^{Σ} results in 0.

$$K^{\Sigma} = \sum_{i=1}^{n} k_i + \sum_{j=1}^{m} u_j \tag{2.1}$$

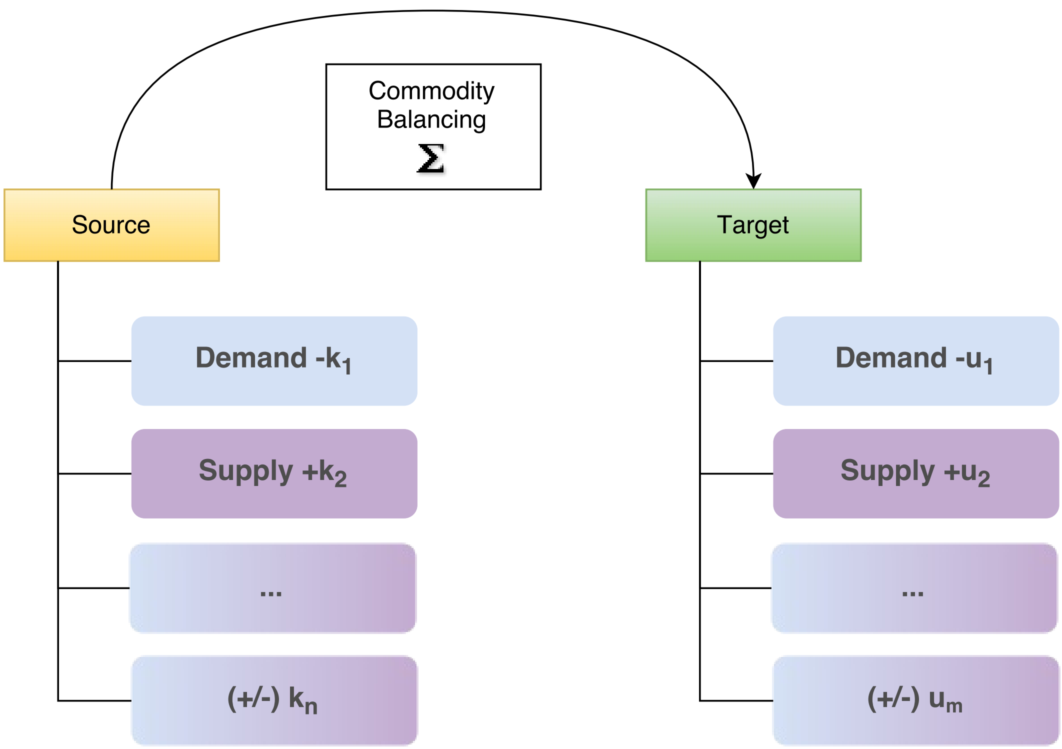

Figure 2.4: Commodity balancing opposes n delivery and m supply amounts so that a trader can balance known demands with the decision variables [BD16].

Example 2.5.1: Consider that a trader has access to a single market area and manages one consumer. Another trader demands 100 units at a certain hour. The deal is operated via the virtual trading point. Furthermore historic consumption data estimates 20 units of gas for the consumer. Supplying amounts are not known, yet, as the upper commodity balance in figure 2.5 demonstrates.
The trader concludes two supply contracts with other traders, which ensure supplies of 80 and 40 units, respectively. Like the delivery contract the supplies are operated via the virtual trading point. The commodity can be calculated as the following:

$$K^{\Sigma} = -100 - 20 + 80 + 40 = 0 \tag{2.2}$$

The equation 2.2 results in 0 and the network will be balanced. In other words, no imbalance costs will be charged. The lower commodity balance in figure 2.5 lists the decision variables and presents a balance between source and target.

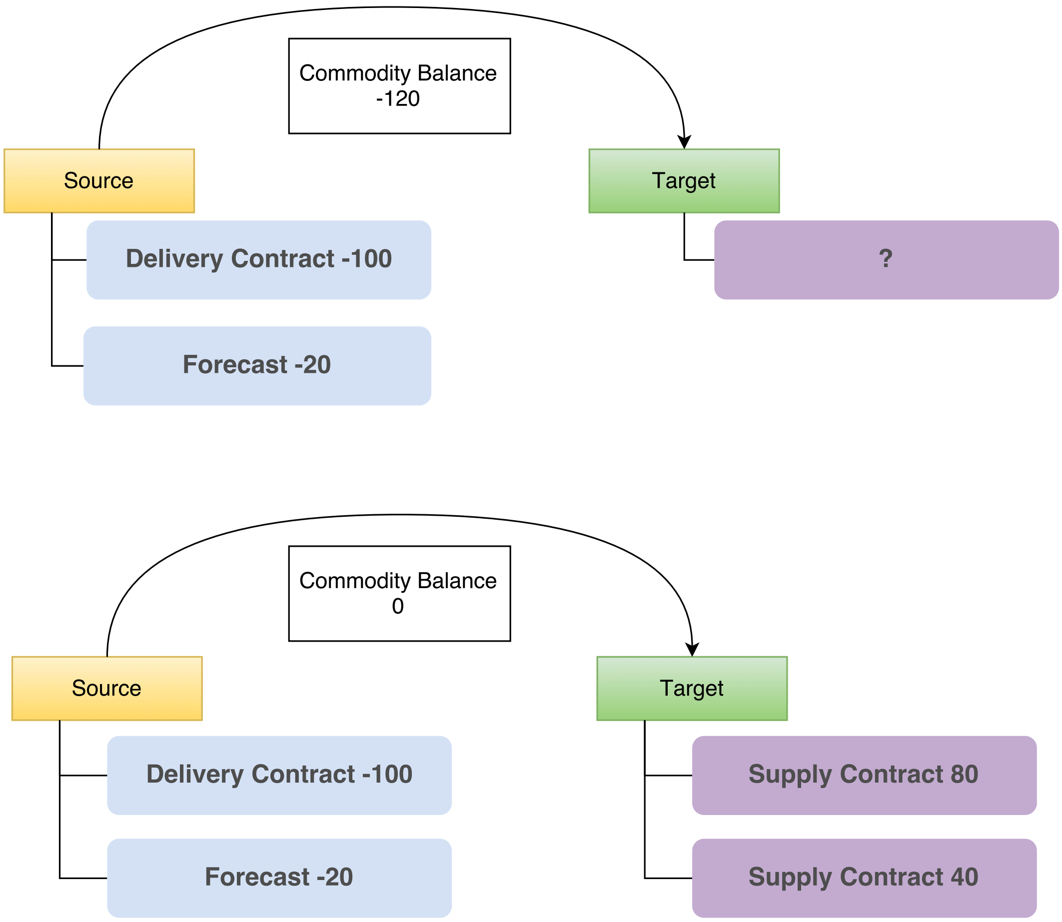

Figure 2.5: Example scenario of a commodity balance. By adding both supply contracts to the target side, the commodity balance can be net out.

The last example 2.5.1 assumes that the trader acts in only one market area. Another constellation occurs if a natural gas trader operates in two adjacent countries:
A trader acts in two market areas m_1 and m_2, as figure 2.6 sketches. The demands of m_2 overbalance the supplies. There are two options, he can try to find attractive options at virtual point 2 or he can try to balance the network by supplying gas via market area m_1. The noted demand and supply amounts and capacities are example values, which are used in example 2.5.2.

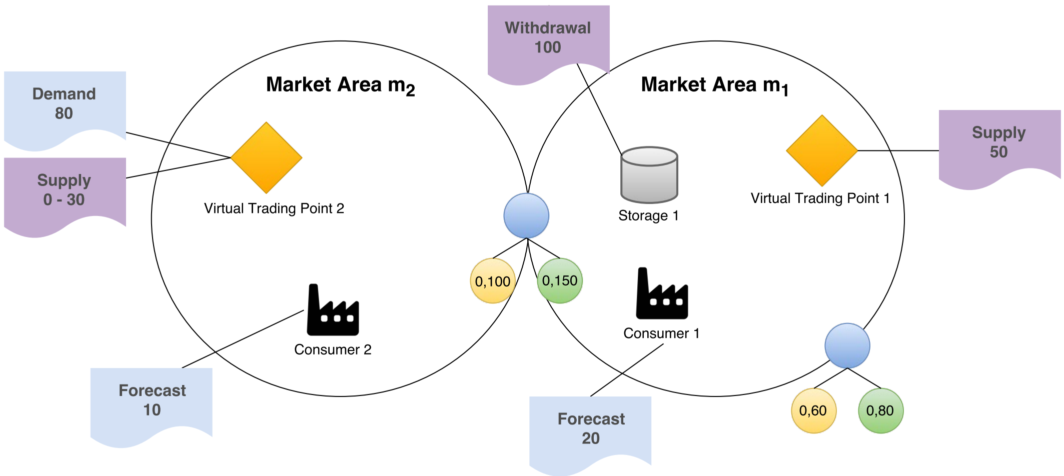

Figure 2.6: A simplified scenario when a trader has access to two adjacent market areas.

Suppose that a trader notices imbalances in market area m_1 and the decision variables in this market area are cost-intensive. If he is able to find attractive supply contracts in market area m_2, he can transfer these to m_1. Consequently he nets out the amounts across commodity balances c_1 and c_2. Equation 2.3 denotes a commodity balancing for multiple market areas.

$$K^{\Sigma} = \sum_{m=1}^{M} \Big(\sum_{i=1}^{K} k_{mi} + \sum_{j=1}^{U} u_{mj} \Big) \tag{2.3}$$

For each market area the single commodity balances are summarized and results in the total commodity balance K^{Σ}.

Example 2.5.2: Consider that a trader is responsible for two market areas m_1 and m_2 as shown in figure 2.6. All sources (delivery demands) and some targets/decision variables (supplies) are known. At the borders of market areas there is a grid point with entry and exit capacities. Assume for simplification that the capacities will be satisfied. Each market area contains a virtual trading point.

The trader has concluded delivery contracts with large-scale consumers in each market area and he has concluded a storage contract in market area m_1.
Furthermore capacities at the border grid points are set. In this example the gas flow is in direction of balancing group II. Market area m_2 shows the following sources and targets:

- There is an amount registration of 80 units via virtual trading point 2
- Another estimation yields 10 units for consumer 2
- A given supply contract via virtual trading point 2 yields a margin between 0 to 30 units of gas.

As mentioned the sources over-weigh the target. Remark that the target is flexible with a margin between 0 and 30 units. Assume that the trader will purchase the maximum amount and there are no further attractive (less cost-intensive) supply sources. The given data for market area m_1

- The trader decides to withdraw 100 units out of storage 1
- An estimation yields 20 units of gas for consumer 1
- There is a fortunate supply contract with 30 units which can be operated via virtual trading point 1

Figure 2.7 exemplifies the commodity balancing for both market areas. Commodity balance c_2 represents market area m_2 and c_1 represents m_1. The trader finds an attractive supply contract via virtual trading point 1 with 30 volume units. The transfer to balance c_2 is done as the following:
A target called *Entry* is defined with the required amount of 60 units in c_2 . This target is associated with a source *Exit* in c_1. The trader balances c_2. The imbalance in c_1 is net out by adding the supply contract to the targets in c_1. As a result both commodity balances are balanced.

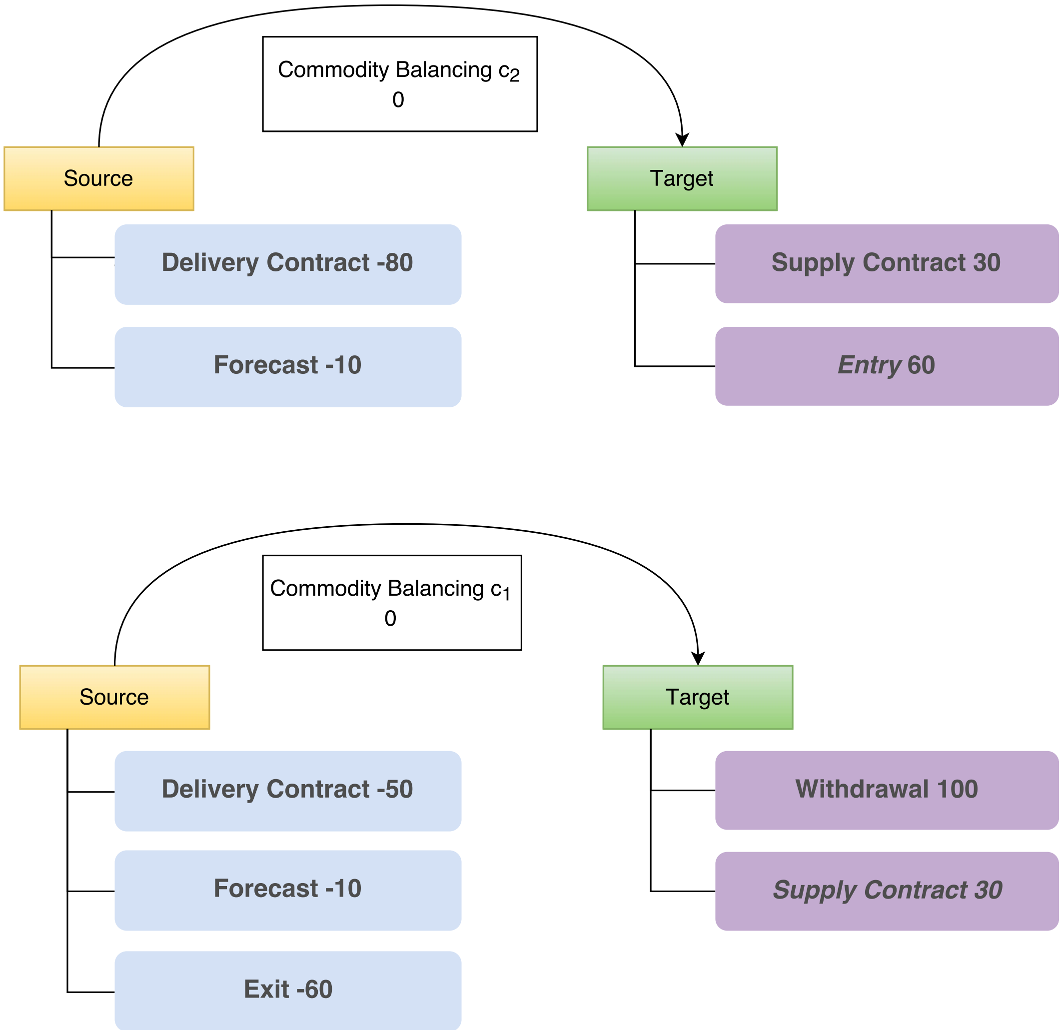

Figure 2.7: A processed commodity balance for the market areas m_2 and m_1. Both balances refer to example 2.5.2 and figure 2.6. Commodity balance c_2 represents market area m_2 and c_1 represents m_1. Target *Entry* in c_2 is the transfer and is associated with source *Exit* in c_1.

Commodity balancing is an important constraint for the trader, since imbalances will be penalized and raise costs. Therefore a trader tries to net out delivery contracts with supply contracts. For a cost-optimal selection there are further constraints that need to be regarded by a trader.

2.6 Further Constraints

A trader has to select the contracts (decision variables) for the day-ahead planning even a shorter time span every day. These contracts have to be picked carefully so that he minimizes costs and earns profit. The problem is to find a good distribution are subject to several contractual conditions, which may be penalized if those are violated. The following constraints influence the selection of decision variables.
Dependencies of former decisions: For simplification it has been assumed that during the day-ahead/intra-day planning phase the single decisions of contracts are independent so that the selection of contracts at a moment t_n does not affect the selection at moment t_{n+1}. Note that different supply options will arrive during the whole day and the trader has to rethink about his distribution.
Figure 2.8 illustrates the different possible supply contracts. In this example a trader has to satisfy 800 units of gas. Assume the day is separated into the moments t_1, t_2, t_3.
At any phase different supply contracts can be chosen by the trader. Each contract constitutes a cost value, for simplicity represented by constant values. First he can either choose two supply contracts with 500 (red path) or 600 (blue path) quantity units respectively. The costs are set to 3 cost units and two cost units. The choice at moment t_1 affects further decisions at later moments and those can depend on contractual conditions or strategic alignment of the trader.

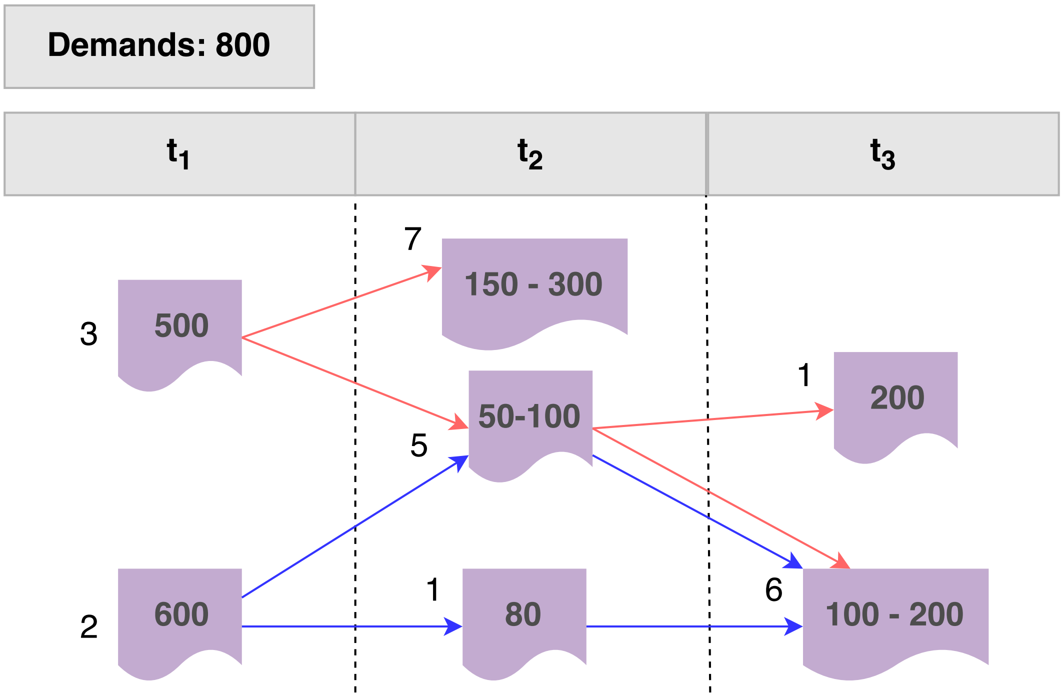

Figure 2.8: Decision options for a trader during a day within three phases. The colours of the arrows indicate the two cases which start with a different amount.

- Case Red Path: The trader has chosen a supply contract of 500 quantity units at moment t_1. If the contract with a flexible margin of 150 to 300 units is more attractive, the trader can choose this contract at moment t_2 and the commodity balance will be balanced. Otherwise he can select the contract with a variable amount between 50 and 100. At moment t_3 he can either choose a contract with a fixed amount of 200 or the variable contract.
- Case Blue Path: Starting from contract 600 the trader can choose either the fixed contract supplying 80 units at moment t_2. Alternatively he is able to select the variable contract with a margin of 50 and 100 units. In both cases the trader will select the variable contract the flexible contract in phase t_3 to balance the demands with the supplies.

Note that a trader ponders over his decisions and redistributes contracts eventually until the daily planning phase has ended. Each path generates dependencies between the contracts but the trader has to decide on the *best* selection of contracts.
Contractual amount limits are defined in delivery or supply amounts. There can be either fixed limits with a constant quantity or flexible limits which define a margin of minimum and maximum limit. Between these limits he is able to determine the gas amount to deliver or purchase. As mentioned the amounts affect the decision of

contracts because a trader tries to balance the gas network (see subsection 2.5).
A further constraint are **capacities**, which are offered by TSOs at a physical grid point. The trader needs to book sufficient capacities in order to provide a demanded gas amount.
Pricing Models are defined as a contractual condition and set a constraint, too. There are several drivers that influence the pricing, e.g. gas production levels, net imports and exports, storage levels, the price of substitutes (oil or coal) or the temperature [Sai16].
Each contract contains therefore an own pricing model which contains dependencies. In most cases the aspects are modelled in the contractual pricing between trading partners. Popular pricing models are fixed pricing models, indexed pricing models and tranche pricing models [ZS09, Bec16].
Fixed pricing models set a constant pricing for a certain amount of gas before the dispatching periods. This price is valid [ZS09] for the defined periods. Natural gas market prices are disregarded, which means that the trader is independent of price variations. As a consequence, the risks are reduced. Since this price is fixed, it can be assumed that the pricing model is linear.
A further model is the natural gas- or oil-indexed pricing model, where the price determination proceeds through three phases: pricing period, time lag and price validity period. During the pricing period, a moving average over historic data (3, 6 or 9 months in the past) is calculated. If a time lag is set, the determined price does not change, e.g. one or three months. In the last period the determined price is charged over a time span, for example three months [Bec16, ZS09].
Another price model can be determined by dividing the total volume into n tranches. There are two periods, a pre-defined order period and a price validity period. During the order period a trader observes the prices and sets them for each tranche. After the order period the price validity period starts [ZS09, Bec16].
For all pricing models there is a long time span before the supply time. In this pricing model it is assumed that the supply period is during the day-ahead planning. A simplified assumption is considering the pricing as a linear function $p(x)$, as shown in figure 2.9. x denotes the gas amount and it is multiplied by a pricing factor m. Fixed costs are denoted as b. Attributes of this price function are that the costs rise monotonically with the gas amount and that $p(x)$ yields a positive value.

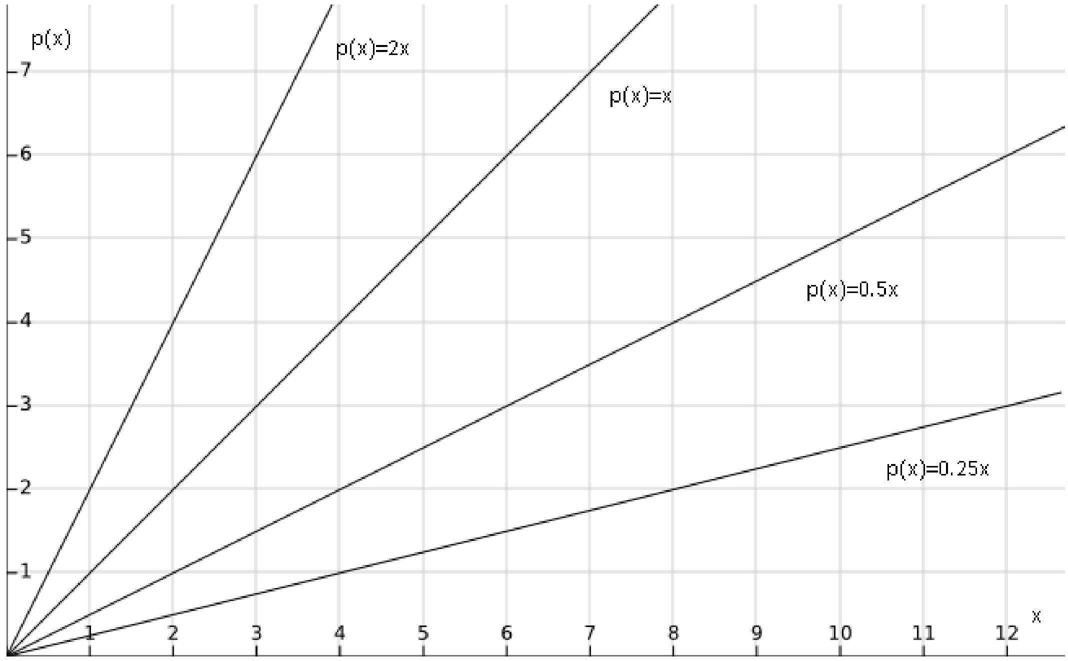

Figure 2.9: Assumed pricing model is linear, i.e. $p(x) = mx + b$

This chapter has outlined the problem of selecting the most economic distribution of supply contracts in the daily planning from a natural gas trader's point of view. It has been discussed that a trader comprises his portfolio with short term contracts for the daily planning and that the trader separates different planning phases.
Specific terminology, the changes at the gas market, contractual parameters and the stakeholders have been discussed. These explanations are important to clarify regulations and constraints which affect the daily selection of the best supply contracts. The fact that a trader searches for a most economic contract distribution raises up the question if this problem is solvable by formulating a linear programming problem. This linear optimization problem needs to be solved such that the trader receives an optimal setting of supply contracts and minimized costs. These results may help the trader to decide for reasonable supply contract distributions.

Chapter 3

A Linear Programming approach to solve Day-Ahead Balancing Problems

There are different motivations of selecting gas supply sources of a trader's given portfolio for the following day. This depends mainly on the natural gas traders primary goals or strategies. One incentive could be the daily balance of the gas network. According to this motivation a trader tries to net out imbalances by operating with long-term, mid-term or short-term contracts. Costs are not considered in the balancing strategy, so far.
Another goal of a trader is to find a cost-minimal distribution with important subgoals, for instance avoiding imbalances and contractual bounds. The different contract types provide various conditions, which need to be regarded for a cost-optimal operation of contracts. A manual determination would be quite difficult, due to the high amount of different parameters. Therefore it is desirable that essential parameters of different contracts are abstracted and passed to a decision making linear programming problem, which helps to find a cost-minimal distribution of a given portfolio.
This chapter deals with the development of a linear program for a cost-minimal selection of a gas traders portfolio. A model that is derived from the following scenario in section 3.1 and the problem description in chapter 2.Furthermore, the principles of linear programming and mixed-integer linear programming are introduced. Afterwards assumptions for the given problem are made, the parameters and variables, constraints and the linear objective function are defined.

These form the main parts of a basic linear optimization problem, which shall exemplify the application of finding a cost-minimal contract selection. In order to approach the problem description, the linear program is extended for further use-cases in the next chapter 4.

3.1 Scenario

A gas trader has concluded a balancing group contract with a market area manager, so that he has access to the market area. Moreover, the trader has concluded a delivery contract with a single consumer. All demand data for the consumer is provided by the market area manager. All trades are operated via the virtual trading point, i.e. there are no capacity bookings.
During a day, the trader receives demands by consumers and other traders or he has available supply contracts over a planning horizon with T different phases. These known sources are elements of the set K. The summary of all sources contractual gas amounts are collected in K^{Σ}.
The primary goal of the natural gas trader is to save costs. To net out K^{Σ} the trader needs to plan with long-term, mid-term or short-term target contracts, which are elements of the set U. For each contract, the trader needs to determine the operational gas amounts. The planned amount to operate with a target contract u is defined by the decision variables x_{tu} where x is the taken volume of contract u in phase t.
In addition, regard that all contracts have a lower and upper bound L_u and U_u, respectively. These bounds define also an interval of available gas amounts for contract u. The pricing $p_{tu}(x)$ determines a linear pricing for contract u in phase t. The trader's goal is to determine a proper selection of gas amounts x_{tu} such that his total costs Z are minimized.

3.2 Background

Allocating contractual amounts of natural gas with the primary goal of minimizing costs can be classified as an optimization problem. This problem type belong to the mathematical field of OR [HL01].
The steps to solve optimization problems are analysing the given real-world problem, formulating a mathematical model and deriving a solution by a mathematical programming algorithm which outputs an optimal value, e.g. minimized costs or maximized profits [HL01, Wil13, Mee13].

These steps are necessary in order to formulate a linear programming problem for the natural gas trader's contract selection.
This section covers the background and characteristics of linear programming (LP). The common terminology, properties and well-known algorithms to solve a linear program are described in section 3.2.1.
In addition, mixed-integer Programming (MIP) is depicted in section 3.2.2. MIPs are specialized linear programs, which cover further assumptions for the underlying mathematical model. Similarly to background section of linear programming the principles and properties are explained and algorithms are briefly outlined. Mixed-integer linear programming plays an important role for some extensions of mathematical model in chapter 4.

3.2.1 Linear Programming

The essence of linear programming is the mathematical formulation which denotes a real-world problem [HL01, Mee13]. Linear programming assumes that a given problem, like scenario 3.1, can be modelled as a linear function, which is either minimized or maximized. Another terminology of the linear objective function is **objective function**, denoted in this work as Z. This function depends on its **decision variables** which are unknown and whose specification determine the optimal output of the objective function [HL01, Mee13].
One exemplary goal (of a natural gas trader) could be: Minimize the costs by selecting appropriate amounts out of n given contracts. The objective function Z represents the total costs while the unknown amounts are denoted as the decision variables x_i. These variables can adopt any continuous value. Remember that the trader has to pay for the taken gas amounts and that these amounts are bound to a price coefficient c_i. This coefficient is called **parameter** [HL01, Mee13]. The variables and parameters could be the following:
Let

$Z(\mathbf{x})$ a cost function that should be minimized

i index variable for a contract

n number of contracts

c_i a pricing coefficient

x_i the gas amounts to operate with

With the given assumptions a linear program could be modelled as denoted in equation 3.1.

$$\text{minimize } Z(\mathbf{x}) = c_1x_1 + \cdots + c_nx_n = \sum_{i=1}^{n} c_ix_i \tag{3.1}$$

A linear programming algorithm determines the contribution of each c_ix_i, such that the costs are minimized. So far, the solution would be $-\infty$, because the linear program is **unconstrained** [HL01, Mee13]. In other words, all decision variables will decrease since there is no boundary for these. Thus, there will be no optimal solution. This is not significant for the trader, though.

Since resources are limited, for instance by lower and upper bounds, production or physical capacities, the decision variables of a linear programming problem are restricted. This is done by linear **constraints** which arrange the decision variables so that they cannot adopt any value. Formally, constraints are defined by an available amount of resources b_i and an amount a_i, which determines an amount of consumption of a resource [HL01, Mee13, Wil13]. There are different types of constraints, such as inequality, equality or non-negativity constraints, as equations 3.2a to 3.2d denote.

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b_1 \tag{3.2a}$$

$$\text{or}$$

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b_2 \tag{3.2b}$$

$$\text{or}$$

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b_3 \tag{3.2c}$$

$$\text{or}$$

$$x_i \geq 0, \; \forall i \leq n \tag{3.2d}$$

Constraint 3.2a defines a maximum amount of resources that all decision variables may not exceed, e.g. an upper bound of a gas contract.

The inequality 3.2b describes a minimum amount which may not remain lower, e.g. a lower contractual bound.

A tight constraint is the equality constraint 3.2c. Suppose for example that all contracts have to balance the network. Each contract can contribute a supply amount, such that the network is balanced equally.

The non-negativity constraint is described in the last equation 3.2d. This constraint allows only positive values or 0 for each decision variable x_i. Depending on the linear programming algorithm, the formulation of the constraints varies, but the sense of them is the limitation of available resources [HL01, Mee13].

Generally, linear programming optimizes the objective function and subject to all defined constraints [HL01, Mee13]. If a constraint is violated, a possible solution is invalid. Formally constraints create a convex region which contains the valid solutions of a linear program, as figure 3.1 illustrates. In the simplified example 3.2.1 it can be observed how the constraints form a convex region and restrict the solution space [HL01, Mee13].

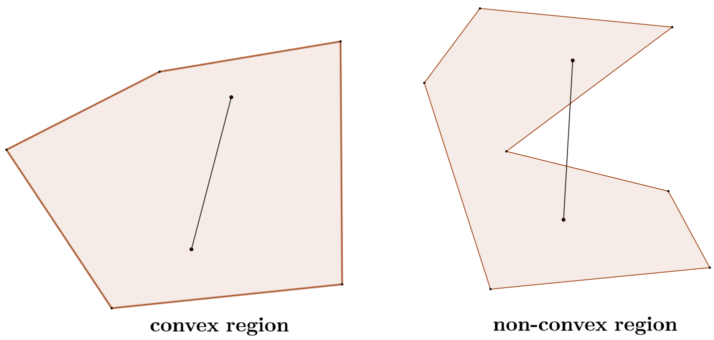

Figure 3.1: Shapes of a convex and non-convex region.

Within the context of linear programming, there are different meanings for the term solution. Any specification of the decision variables x can generally provide a solution of the problem. The following solution terms are delimited [HL01, Mee13].

- **Feasible solutions** satisfy all constraints and these are valid solutions.
- **Infeasible solutions**, however, are invalid because those solutions violate at least one constraint.

A linear program provides an **optimal solution** where the objective function Z has either the largest (in case of maximization) or lowest (in case of minimization) value. Depending on it's definition and parametrization, linear programs can be infeasible, provide a single optimal solution or multiple optimal solutions [HL01]. It is infeasible, each setting of the decision variables violates the defined constraints. If it yields multiple solutions, the objective function Z results in the same value but there is an infinite number of settings for the decision variables x [HL01].
To illustrate the described principles, example 3.2.1 shows a graphical solution approach for a simple linear program.

Example 3.2.1: Let 3.3 denote a maximization problem subject to it's constraints.

$$
\begin{aligned}
&\text{maximize } Z(x_1, x_2) = \left(\frac{1}{2}x_1 + x_2\right) \\
&\text{subject to} \\
&\qquad x_1 \leq 4 \\
&\qquad x_2 \leq 6 \\
&\qquad 2x_1 + x_2 \leq 12 \\
&\qquad x_1, x_2 \geq 0
\end{aligned}
\tag{3.3}
$$

Figure 3.2 shows the graphical solution of the given example. As one can see, the constraints create a convex region, which contains all feasible solutions. All values inside this region satisfy the constraints in equation 3.3. To solve the mathematical programming problem graphically, first, a value for Z is guessed, e.g. $Z(x_1, x_2) = 3$ which leads to equation for a line denoted by $x_2 = -0.5x_1 + 3$. This line can be shifted until it reaches the extreme point, the corner point that results to the maximum value of the objective function.
Point $(3, 6)$ looks promising. By inserting the values $x_1 = 3$ and $x_2 = 6$ into our objective function and obtain $Z(3, 6) = 7.5$. Since the corner point is still part of the convex region, all constraints are satisfied and from this it follows that the solution is feasible. This can be proved by inserting the values into each constraint.

A graphical solution approach is possible only if the optimization problem has two or at most three decision variables. Most real world problems, like the natural gas trader's day ahead balancing problems consist of more than three decision variables, though.
Several algorithms have been developed in order to solve more complex linear programming problems. One of the most popular is the simplex algorithm by Dantzig [Dan63], which is implemented in most open-source and proprietary solvers. For most problems this algorithm finds efficiently a global optimum. Several modifications and extensions have been made to improve the simplex algorithm. Further algorithms, for example Karmarkar's algorithm which is an interior point technique to solve a linear program [Kar84].

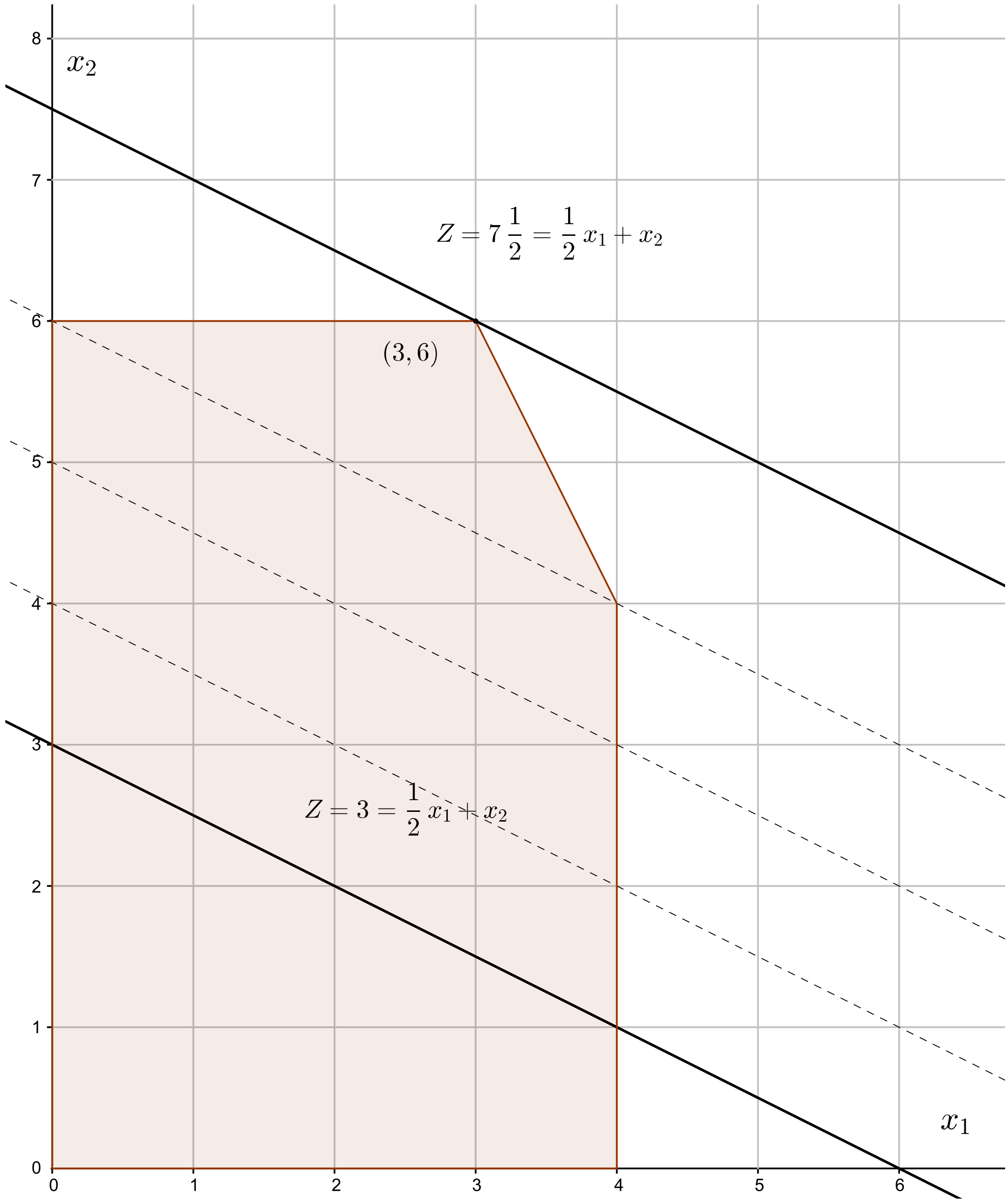

Figure 3.2: Graphical approach to solve a linear program.

To conclude this section, linear programming optimizes an objective function subject to its constraints to plan with amounts of limited resources. If the linear optimization problem results in a feasible solution, an optimal setting of all decision variables is found. While this section has introduced the basic terminology and an abstract sample, the foundations of linear programming are applied in section 3.3, which deals with the gas trader's day ahead balancing problem. In chapter 4 which covers modifications for further problem cases, linear programming comes to its limits.

One extension, for example, are take-or-pay clauses which are denoted as a yes-or-no decision. In order to formulate such yes-or-no decisions, a pure linear programming model has to regard further restrictions, e.g. integral values for some variables. Mixed-integer programming is suitable for these purposes.

3.2.2 Mixed-Integer Linear Programming

In general, Linear programming does not make assumptions about the domain of decision variables. The simplex algorithm, for example, assigns continuous values to the decision variables. For some problems it is desired that the decision variables adopt integer values [Wil13, HL01]. Consider, for example, that a linear program should decide about the amount of products to sell, warehouse stockings, or people are assigned to a team, which have to be indivisible.
Moreover it is hard to model logical conditions because linear programming decides about amounts while it does not make decisions about cases. These planning problems can be formulated with integer programming where integer values can be assigned to decision variables [Wil13, HL01].
This section covers the applicability of integer programming and briefly introduces algorithms which solve integer programming problems.
In the common literature there are different types of integer programming:

- If all variables are restricted to be integer, then it is called *pure integer programming* (PIP) [HL01, Wil13].
- If there are only binary values allowed, then the optimization problem is called *binary integer programming* (BIP) [HL01, Wil13].
- Mixed-Integer (Linear) Programming (MIP) allows that there are both, integer values and continuous values.

One way to allow integer values is to solve the problem with by a linear program (with continuous values) and round them afterwards. This works well if the decision variables adopt rather large values and the resulting error is small. One drawback appears in smaller problems. A provided solution may not be feasible after rounding and it is hard to see in which way the optimization algorithm shall round to retain feasible solutions. Example 3.2.2 shows that rounding can yield infeasible solutions for an integer programming problem [HL01].

Example 3.2.2: Let 3.4 denote a maximization problem [HL01]:

$$
\begin{aligned}
&\text{maximize } Z(x_1, x_2) = x_2 \\
&\text{subject to} \\
&\quad -x_1 + x_2 \leq \frac{1}{2} \\
&\quad x_1 + x_2 \leq \frac{7}{2} \\
&\quad x_1, x_2 \geq 0 \text{ and } x_1, x_2 \text{ are integers}
\end{aligned} \tag{3.4}
$$

The obtained solutions will be $x_1 = \frac{3}{2}$ and $x_2 = 2$. As figure 3.3 pictures rounding leads to infeasible solutions because both $x_1 = 1$ and $x_1 = 2$ are not in the convex region, and the result for a linear program could be erroneous.

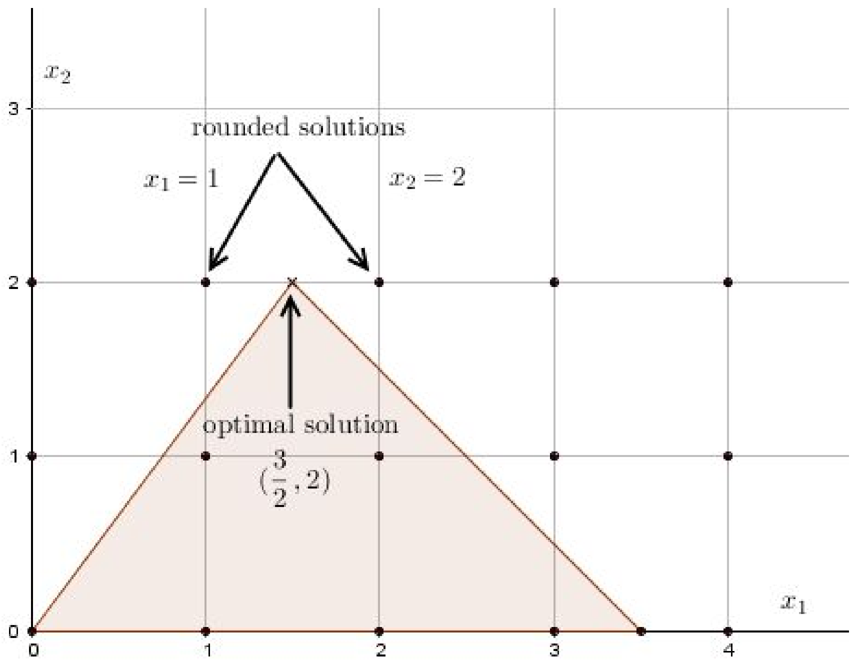

Figure 3.3: Rounding the optimal solution may yield infeasible solutions [HL01].

Another problem is that even if an algorithm yields a feasible solution, there is no guarantee that the rounding leads to the optimal solution [HL01]. This is shown by example 3.2.3.

Example 3.2.3: Let 3.5 denote another maximization problem [HL01]:

$$\begin{aligned} &\text{maximize } Z(x_1, x_2) = x_1 + 5x_2 \\ &\text{subject to} \\ &\qquad x_1 + 10x_2 \leq 20 \\ &\qquad x_1 \leq 2 \\ &\qquad x_1, x_2 \geq 0 \text{ and } x_1, x_2 \text{ are integers} \end{aligned} \tag{3.5}$$

Figure 3.4 depicts non-optimality of rounding afterwards. Linear solvers would set $x_1 = 2$ and $x_2 = \frac{9}{5}$ and the optimal solution $Z = 11$. Rounding towards the feasible region would produce $x_1 = 2$ and $x_2 = 1$ which yields $Z = 7$, but this solution is not the optimal solution. Assign $x_1 = 0$ and $x_2 = 2$ yields $Z = 10$, which is a feasible solution for this optimization problem.

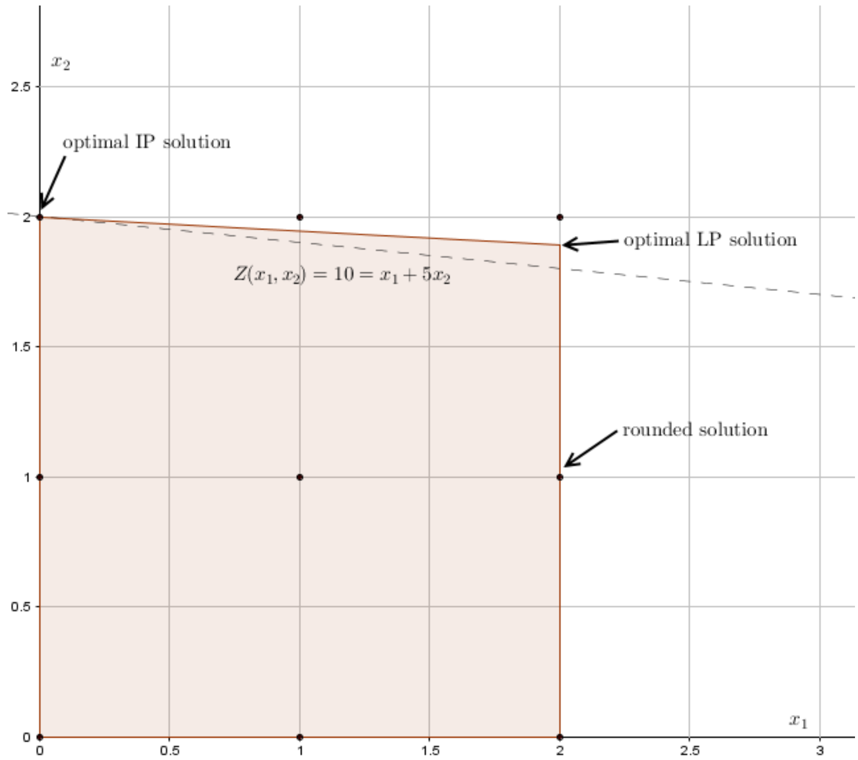

Figure 3.4: Rounding does not always lead to an optimal solution. [HL01].

Because of these two pitfalls, sophisticated algorithms have been developed in order to solve integer programming problems [HL01, Wil13]. One popular algorithm is the branch-and-bound algorithm. First a given integer programming problem is solved like a linear program by the simplex algorithm. The integrality constraints

are relaxed. If the solution is an integer value, the problem is solved, otherwise a tree search is performed [Wil13].
Another well known algorithm is the cutting planes method. Similarly to the branch-and-bound algorithm, the given integer problem is solved by a linear solver with relaxed integrality constraints. If the obtained solution is an integer, the algorithm will stop. Otherwise the integrality constraints are systematically added to the problem until the integral constraints are satisfied [Wil13].
MIP offers the formulation of logical conditions, which are hard to model solely by linear programming. Logical conditions can be realised by introducing binary variables. These variables can either be implemented in the objective function or the linear constraints to regard yes-or-no decisions [Wil13, HL01].
The following example 3.2.4 demonstrates how an either-or-decision can be implemented for a linear program.

Example 3.2.4: Suppose a natural gas trader has concluded two supply contracts but he is able to operate with only one contract per day and he has to net out 2 energy units. Contract 1 offers natural gas in the interval $[1,2]$ and contract 2 offers $[2,3]$. The pricing factor c_1 and c_2 are multiplied with the natural gas amount. A linear minimization problem can be formulated as equation 3.6

$$\begin{aligned}
&\text{minimize } Z(x_1, x_2) = c_1x_1 + c_2x_2 \\
&\text{subject to} \\
&\qquad x_1 + x_2 = 2 \\
&\qquad 1 \leq x_1 \leq 2 \\
&\qquad 2 \leq x_2 \leq 3
\end{aligned} \tag{3.6}$$

The problem is that x_1 and x_2 are forced to take a value greater than 0 because of the contractual bounds. As a consequence, the first constraint cannot be satisfied and the linear program yields an infeasible solution. To dissolve this lack either contract 1 or contract 2 has to be deactivated. This can be achieved by introducing a binary variable y and by modifying the boundary constraints as the following:

$$\begin{aligned}
y \leq x_1 \leq 2y \\
2(1-y) \leq x_2 \leq 3(1-y) \\
y \in 0, 1
\end{aligned} \tag{3.7}$$

If $y = 1$, the boundaries for contract 1 are active and x_1 can adopt a value greater than 0 while the second constraint forces $x_2 = 0$. In case of $y = 0$ contract 2 is active and x_1 is forced to adopt 0.

Example 3.2.4 has shown a simplified either-or-decision by adding binary variables. In some cases, also the objective function may be affected, e.g. if there are fixed charges when a contract is not regarded [HL01].
It has to be mentioned that MIPs are sensitive to the amount of integer variables, while continuous variables only have almost no effect on the computational effort [HL01]. For each added variable the computational effort increases since all combinations of possible variable settings have to be calculated [HL01].

Since simple rounding of the solution could create an infeasible solution or generate a non-optimal solution, the applicability of MIP has been presented in this section. This specialized linear program adds the restriction that some variables of the linear optimization problem can adopt only integer values. Furthermore it is possible to model yes-or-no decisions with MIP, such that constraints can be deactivated.
With the covered background of linear programming and mixed-integer programming, an optimization problem to solve a cost-optimized day-ahead balancing problem is developed in section 3.3 and extended chapter 4.

3.3 Application

Recapitulate that the goal of a natural gas trader is to determine the minimal costs by picking the gas amounts out of the given target contracts. For the first linear model draft, parameters and variables have to be defined and assumptions have to be made [Mee13].
First, it is assumed that the data, e.g. the prices to operate with a contract, is known for the day-ahead planning. This helps to model a linear objective function [Mee13]. In order to approximate the complex problem of the day-ahead balancing problem, a simplified linear program is proposed in this section. The objective function of this problem is minimized such that a cost-optimal distribution is yielded.

3.3.1 Parameters and Assumptions

First of all, the linear mathematical model and constraints need to be formulated [HL01, Mee13]. On beforehand the notations are clarified and some assumptions are made for the given scenario 3.1.

Let

- T Planning horizon, e.g. a day with 24 phases. Phase $t = 1$ is the start of the day
- t Concrete phase of the day
- K Set of all known contractual gas amounts which are listed at the source side of the commodity balance 2
- k Concrete source side contract, where $k \in K$
- K^{Σ} Sum of all sources which is a numerical value, let $K^{\Sigma} = \sum_{k \in K} k$
- U Set of all available target contracts, listed at the target side of the commodity balance 2
- u Concrete target contract, where $u \in U$
- L_u Contractual lower bound of contract u, such that $L_u \leq x_{tu}$
- U_u Contractual upper bound of contract u, such that $x_{tu} \leq U_u$
- $p_{tu}(x)$ Contractual pricing model which is determined by phase t and contract u
- m_{tu} Pricing factor at phase t for contract u
- x_{tu} Selected amount of natural gas with contract u at phase t (decision variable). All gas amounts are real numbers, $\forall x_{tu} \in \mathbb{R}$
- $Z(\mathbf{x})$ the total costs (objective function) depending on all taken gas amounts $\mathbf{x}$

Note that some parameters, for instance the pricing function, depend on both, a certain contract u and a phase t. The contractual bounds are valid for the whole planning horizon T. In other words, a trader can only operate with a daily amount of gas. For the linear program and its given parameters further assumptions are made:

- At the end of the planning horizon T, the sum of all x_{tu} shall net out K^{Σ}
- The contractual limits L_u and U_u define an interval of available gas amounts.
- There are no storage costs and no physical capacity costs.
- Suppose here that there are no fixed costs if no gas is taken.
- All pricing models are linear, such that a linear program can be applied.

The last assumption is very particular because the gas market is volatile [ZS09]. Here it is assumed that all contractual pricing conditions are defined in a factor m_{tu}, because the trader already knows about all price circumstances for the next day 2. All pricing functions define a linear function, as figure 2.9 sketches. Hence the costs to operate with a contract u in phase t is calculated by the product $m_{tu}x_{tu}$. As long as the pricing model $p_{tu}(x)$ is a linear function, the optimization problem is solvable for a linear program.
All given parameters are used for the formulation of linear constraints in section 3.3.2 and the objective function 3.3.3 respectively.

3.3.2 Constraints

According to the problem analysis there are constraints which specifies (or limit) the values of every decision variable 2. There are two important constraints in the first approach, the balance and contractual bounds.
It has to be ensured that the decision variables net out the daily demand K^{Σ} because a trader wants to avoid imbalances [ZS09]. More formally, the total sum of all gas amounts of all available target contracts over the whole planning horizon T are equal to K^{Σ}, as denoted in equation 3.8.

$$\sum_{t=1}^{T} \sum_{u \in U} x_{tu} = K^{\Sigma} \tag{3.8}$$

Another important constraint is the observance of contractual bounds over the whole planning horizon T. This means that a further constraint has to ensure that the natural gas amounts are in between their contractual bounds L_u and U_u. This means that two further constraints complete the linear program.
Equation 3.9 formalises both constraints in a single line. The first constraint regards that the sum of operated natural gas of a single contract is greater than or equal it's lower bound L_u. The same sum has to be lower than or equals the upper bound U_u.

$$L_u \leq \sum_{t=1}^{T} x_{tu} \leq U_u, \forall u \in U \tag{3.9}$$

All constraints ensure that x_{tu} is limited and reduce the solution space of the linear program. In the subsequent step the objective function is drafted, which shall be minimized.

3.3.3 Objective

The linear program needs to determine the values for the decision variables x_{tu} such that the costs are minimized. The total cost function $Z(\mathbf{x})$ is defined by the sum of all contractual pricing models $p_{tu}(x)$ with the taken gas amounts x_{tu} over all phases of the planning horizon T.

$$\text{minimize } Z(\mathbf{x}) = \sum_{t=1}^{T} \sum_{u \in U} \Big(p_{tu}(x_{tu}) \Big) \tag{3.10}$$

Equation 3.10 denotes the linear model for the objective function of the linear program. Combined with constraints 3.8 and 3.9 the complete simplified propose of the linear program is denoted as equation 3.11:

$$\begin{aligned} &\text{minimize } Z(\mathbf{x}) = \sum_{t=1}^{T} \sum_{u \in U} \Big(p_{tu}(x_{tu}) \Big) \\ &\text{subject to} \\ &\qquad \sum_{t=1}^{T} \sum_{u \in U} x_{tu} = K^{\Sigma} \\ &\qquad L_u \leq \sum_{t=1}^{T} x_{tu} \leq U_u, \forall u \in U \end{aligned} \tag{3.11}$$

The linear program of equation 3.11 minimizes the total operational costs by looking for the best specification of all amounts x_{tu}.
This fundamental linear program is modified by adding further contractual conditions or other contract types, for instance storage contracts, in section 4.
On beforehand, example 3.3.1 shall show if it is already possible to obtain an answer for the formulated optimization problem. Suppose that the inserted numbers are only for testing. In reality, the operational gas amounts are a multiple thereof.

Example 3.3.1: First, it is assumed that the day is separated in two planning phases, for example day and night. A natural gas trader has to balance a total demand of 3 energy units and he wants to operate with three different target contracts, which offer flexible contractual bounds.

Let

$$
\begin{aligned}
T &= 2 \\
U &= (\text{Flex}_1, \text{Flex}_2, \text{Flex}_3) \\
K^{\Sigma} &= 3
\end{aligned}
$$

Secondly, all contracts define flexible bounds, where x_{tu} has to be in between the interval $[L_u, U_u]$. The bounds are valid for the whole day, so there are no phase-dependent bounds:

$$
\begin{aligned}
\text{Flex}_1 &= [1,2] \\
\text{Flex}_2 &= [0,2] \\
\text{Flex}_3 &= [0,1]
\end{aligned}
$$

The pricing models for the given contracts are defined as linear functions $p_{tu}(x) = m_{tu}x$. m_{tu} denotes a contractual pricing factor for contract u in phase t. Table 3.1 lists the specification for m_{tu}

U	t	m_{tu}
$Flex_1$	1	4
	2	5
$Flex_2$	1	2
	2	1
$Flex_3$	1	1
	2	3

Table 3.1: Price factor setting of the example application.

All parameters can be inserted into the linear program, which creates the concrete optimization problem 3.12:

$$\begin{aligned}
&\text{minimize } Z(\mathbf{x}) = \Big(4x_{11} + 2x_{12} + x_{13}\Big) + \Big(5x_{21} + x_{22} + 3x_{23}\Big)\\
&\text{subject to}\\
&\qquad \Big(x_{11} + x_{12} + x_{13}\Big) + \Big(x_{21} + x_{22} + x_{23}\Big) = 3\\
&\qquad 1 \leq (x_{11} + x_{21}) \leq 2\\
&\qquad 0 \leq (x_{12} + x_{22}) \leq 2\\
&\qquad 0 \leq (x_{13} + x_{23}) \leq 1
\end{aligned} \tag{3.12}$$

To solve this problem computationally, the Python-based optimization modelling language Pyomo in combination with GNU Linear Programming Kit (GLPK) [HWW11, Inc12]. The obtained costs for this example is $Z(\mathbf{x}) = 6$ cost units for the contract setting $x_{11} = 1$, $x_{13} = 1$, $x_{22} = 1$. Table 3.2 lists the setting subject to contract and phase.

U	t	x_{tu}	$p_{tu}(x)$
$Flex_1$	1	1	4
	2	0	0
$Flex_2$	1	0	0
	2	1	1
$Flex_3$	1	1	1
	2	0	0

Table 3.2: Result setting for all x_{tu} with an optimal cost of $Z(\mathbf{x}) = 6$

Consider that the lower bound of contract $Flex_1$ forces the linear program to select at least 1 energy unit of contract $Flex_1$ in phase 1. Thus 4 cost units for $x_{11} = 1$ are added to the total costs. For each the contributions of x_{22} and x_{13} only 1 cost unit is added respectively. Consider that it would be cheaper to skip contract $Flex_1$ and operate with either $Flex_2$ in phase 2 or with contract $Flex_3$ in phase 1 since no penalty costs are invoiced if contract $Flex_1$ would be disregarded.

This section has covered a basic application of a linear programming problem. The obtained solution for the simple example looks promising, but the tight contractual bounds, as denoted in equation 3.9, forces the linear program to select expensive contracts. This is undesired, because a trader would rather ignore cost-intensive contracts than operating with them.
The following chapter 4 proposes extensions for the model, such that this problem is eliminated and further circumstances are regarded, for example take-or-pay clauses, operating storages or booking further capacities.

Chapter 4

Adaptations of the Linear Programming Problem

The background and basis application of the previous chapter 3 has given a first insight of linear programming. The proposed optimization problem has not been sufficient, though, because there are further assumptions to be made and constraints to regard, as chapter 2 has outlined. To come closer towards a realistic application the basic linear programming problem has to be extended.

The following sections deal with the gradual adaptation of the linear program. Section 4.1 discusses fixed contracts and introduces a binary variable to deactivate expensive ones. The linear programming approach becomes to a mixed-integer programming problem (MIP). Closely connected to this extension are take-or-pay penalties, which generate costs, if a contract is disregarded. This type of penalization is explained in section 4.2.

A special contract type are storages because the trader can either inject or withdraw gas of his available storages. The involved characteristics of natural gas storages are clarified in section 4.3.

By adding daily and hourly imbalance penalties to the mathematical model in the sections 4.4 and 4.5, the optimization problem gets a further cost parameter, which shall be minimized. The last two extensions cover additional capacity costs and market options in the sections 4.6 and 4.7, respectively.

All extensions are concluded to the mixed integer problem in section 4.8. Last but not least, the determination of strategic bounds in a pre-processing step is debated in section 4.9.

4.1 Disregard contracts and Fixed contracts: Yes-or-No Decisions

Suppose that all contracts define a lower and an upper bound as a constraint in the linear optimization problem. The solver of a programming problem decides to operate with a contract or not because if the lower bound L_u is set to zero. The resulting costs for the contract will be zero, as well.
If a lower bound is greater than zero, i.e. $L_u > 0$, then the solver takes at least the value of L_u, even though the costs are higher, because of the tight constraint as denoted in equation 3.9, forces the algorithm to satisfy the constraint.
This behaviour can also occur, if a trader likes to operate with a fixed contract. A fixed contract defines a constant energy amount. Since the hard constraint 3.9 would lead to select a contract every time, the model needs to be extended to activate or deactivate the bounds.
First, if a contract has fixed contractual bounds, the lower and upper bound can be set to be equal, i.e. $L_u = U_u$. For each fixed contract x_{tu} will accept the boundary value, because of constraint 3.9. For each contract u an auxiliary binary variable y_u is introduced, whose accepted values are either 0 or 1 [HL01, Wil13, Bis16, Bis09]. This binary variable indicates if a contract will be regarded or not for the whole planning phase T.
Let

y_u = Decision variable which determines if a contract shall be selected (=1) or not (=0), $y_u \in 0, 1$

More formally the relation between y_u and each x_{tu} is denoted as the following implications:

- $y_u = 0 \implies \sum_{t=1}^{T} x_{tu} = 0, \forall u \in U$
- $y_u = 1 \implies L_u \leq \sum_{t=1}^{T} x_{tu} \leq U_u, \forall u \in U$

These implications can also be denoted as equation 4.1.

$$x_{tu} = \begin{cases} 0, & \text{if } y_u = 0 \\ L_u \leq x_{tu} \leq U_u, & \text{if } y_u = 1 \end{cases} \tag{4.1}$$

In order to implement the relation, the contractual boundary constraint is modified (see 4.2).

$$
\begin{aligned}
& y_u L_u \leq \sum_{t=1}^{T} x_{tu} \leq y_u U_u \\
& y_u \in 0, 1
\end{aligned}
\tag{4.2}
$$

If $y_u = 0$, then x_{tu} is forced to adopt 0. On the contrary, if $y_u = 1$, then both bounds are activated and the decision variable x_{tu} can accept any value in between. The following example shows the extension of the solution space by adding binary variables.

Example 4.1.1: Consider that there is only one contract u and a single planning phase $T = 1$. The contract defines a lower bound $L_u = 1$ and an upper bound $U_u = 5$. Using the binary variable y_u with constraint 4.2 extends in that case the feasible region of a linear program, as figure 4.1 illustrates.

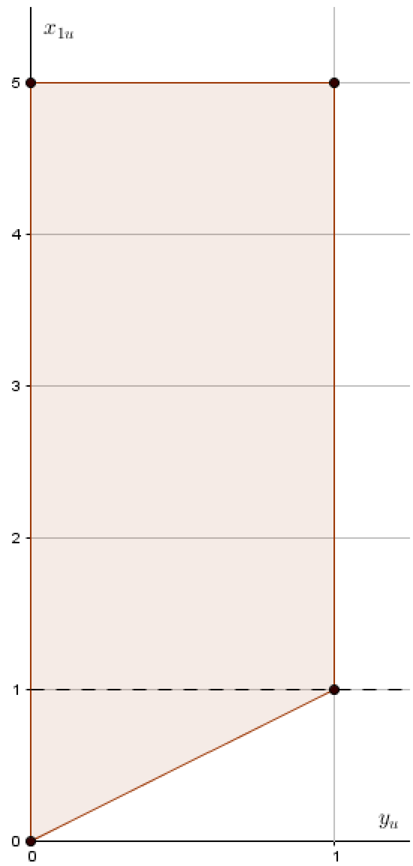

Figure 4.1: Effect of extending the feasible region by the binary variable y_u.

If there were no constraint, the dashed line would be the lower bound of the feasible region and a solving algorithm would set values in between the dashed line and the upper bound. By combining the boundaries with the binary values, the linear solver is able to allow also 0 as value for x_{1u}.

By introducing the binary variable y_u which accepts only values 0 and 1, the linear program becomes a mixed-integer linear programm. The optimization problem is resolvable if the underlying solver is able to optimize MIP [HL01, Wil13]. If so, all variations of the binary variable y_u and the continuous variable x_{tu} are calculated so that the minimal costs can be computed.
A problem that might occur is that the amount of binary variables grows proportional with the amount of contracts. Consider that for each contract u a new binary variable has to be added and in consequence more combinations have to be computed by the solver [HL01, Wil13]. In addition, binary variables play a role for take-or-pay rates, which are introduced in the next section 4.2 in order to invoice fixed costs.

4.2 Take-or-pay: Penalize disregarded contracts

Section 4.1 has already introduced binary decision variables in order to activate or deactivate contracts. Those changes has primarily affected the contractual boundary constraint 3.9. But what happens if a contract contains conditions which penalize a defiance of a contract?
So called take-or-pay clauses are a contractual mechanisms which generate costs if a gas trader disregards a contract, i.e. it is a penalty or compensation which has to be paid to the contractual partner [ZS09]. Therefore take-or-pay clauses have an impact on the choice of contracts. A trader has to weigh up which decision is less cost-intensive: Either disregarding take-or-pay contracts and paying the compensation or operating with the contract.
The given linear model needs to be extended by a fixed cost term [HL01, Wil13]. Suppose that a take-or-pay clause is defined by a *take-or-pay-rate* which accepts continuous values between 0 and 1.
Let

$\gamma(\mathbf{y})$ take-or-pay penalty function which sums up penalties for all contracts.

θ_u The contractual take-or-pay rate, where $0 \leq \theta_u \leq 1$

It is assumed that θ_u is coupled with the lower bound times the lowest pricing factor of all m_{tu}. In other words, if a contract is disregarded the lower bound is invoiced and θ_u controls the charged pricing, e.g. take-or-pay ($\theta_u = 1.0$) or take-or-pay-half ($\theta_u = 0.5$).
In order to implement the fixed costs, equation 4.3 defines the take-or-pay penalty function which calculates take-or-penalty costs for each contract and summarises each cost value in a total cost value.

$$\gamma(\mathbf{y}) = \sum_{u \in U} \Big((1 - y_u) \cdot \theta_u \cdot L_u \cdot \min_{1 \leq t \leq T} (m_{tu}) \Big) \tag{4.3}$$

This function is added to the objective function, as equation 4.4 denotes:

$$\text{minimize } Z(\mathbf{x}, \mathbf{y}) = \underbrace{\sum_{t=1}^{T} \sum_{u \in U} \Big(p_{tu}(x_{tu}) \Big)}_{\text{operational costs}} + \underbrace{\gamma(\mathbf{y})}_{\text{take-or-pay penalty}} \tag{4.4}$$

Even though binary variables, which are denoted by $\mathbf{y}$, are added to the objective function the problem is still linear and therefore the problem is manageable by MIP algorithms [HL01, Wil13]. Remember that the constraint 4.2 controls if a contract shall be regarded ($y_u = 1$) or not ($y_u = 0$), because y_u either activates or deactivates the daily bounds of a contract.
The cases to examine are the following: If y_u equals 0, all x_{tu} of a contract u over the complete planning horizon T will result 0 and therefore the first part $m_{tu}x$ equals 0. The second part will become active, if $\theta_u > 0$.
If, however, $y_u = 1$, x_{tu} will adopt a value in between the bounds, i.e. the first part of the pricing model is active and the second is deactivated and the costs for a supply can be calculated. In addition, when there are no contractual take-or-pay clauses, parameter θ_u can be set to 0 such that the second term cannot become active, although a contract is disregarded by the solver.
Up to here, delivery and supply contracts have been examined and those contracts can be activated or deactivated. A further use-case is the operation of storages where the trader has to decide if gas is withdrawn or injected.

4.3 Storages

Storages take on a particular role because the trader can either decide if he likes to inject gas into a storage, or withdraw gas out of a storage. Consider the case that a trader operates only with supply contracts which are elements of K, the set of known

contracts. In order to balance the network, and satisfy the balance constraint 3.8, the linear programming problem needs to control the injection or withdrawal of natural gas [Hol08]. Each operation results in transportation costs, which are invoiced by a TSO and need to be regarded in the linear programming model.
Figure 4.2 models the structure of a natural gas storage. Together with contractual lower and upper bounds, there are physical bounds, a maximum volume V_{max} and a minimum volume V_{min}. As depicted, latter can vary because a system storage operator may conclude a cushion to maintain pressure in the storage. The current volume V stands for the available amount which is decreased in case of a withdrawal or increased in case of injection.

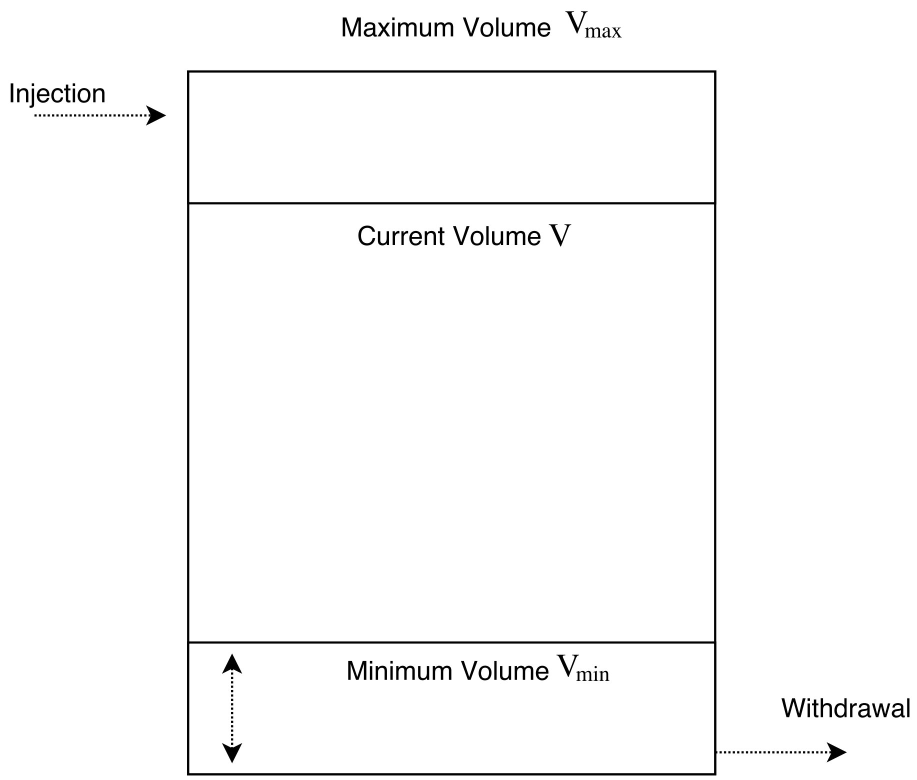

Figure 4.2: Structure of a storage

These physical limits have to be regarded in the linear programming problem. Moreover both storage operations need pricing coefficients, such that the costs for injection or withdrawal can be calculated.
Therefore, let

U_{St} Subset of unknown contracts containing only storages, $U_{St} \subseteq U$

V Current volume inside of the storage

V_{min} Minimum allowed volume of the storage

V_{max} Maximum allowed volume of the storage

m_{tu} Cost factor for withdrawal of gas, $m_{tu} \geq 0$

n_{tu} Cost factor for injection of gas, $n_{tu} \geq 0$

The injection of gas changes the domain of possible adoptable values of the decision variables x_{tu}. In former examples, it has been assumed that all contracts provide supply amounts and thus the decision variables adopt positive values. Remember, that demand contracts are assigned with a negative sign in the commodity balance 2, because the demanded gas leaves the market area grid of a trader. Similarly to this, an injection can also be thought of a demand, which is set by the trader. In order to accept negative x_{tu}, the lower bound L_u can be set negative in case of a storage. As a consequence, the decision variables x_{tu} can also adopt a negative value. In contrast, the withdrawal can be thought of as a supply contract, like in former assumptions and examples. Hence, positive values represent the withdrawal of natural gas, whereas $x_{tu} = 0$ represents that a storage is not used.
A trader has to pay for operational costs. Since negative costs are not allowed in that case, the pricing model p_{tu} contains an absolute function and distinguishes between positive and negative decision variables, as defined in equation 4.5.

$$p_{tu}(x) = \begin{cases} m_{tu}x_{tu}, & \text{if } x \geq 0 \\ n_{tu}|x_{tu}|, & otherwise \end{cases} \tag{4.5}$$

Note here, that the factors $m_{tu} \geq 0$ and $n_{tu} \geq 0$. The absolute value of x_{tu} will ensure that the costs are positive. Figure 4.3 shows a exemplary pricing function.
There is an drastic issue with that pricing model: The absolute function is not linear and thus it is not applicable for a linear program [HL01]. Assuming that the pricing factors are positive and the pricing model is a piecewise linear function, the absolute function can be linearised by adding new constraints [Bis16, Bis09, Wil13].

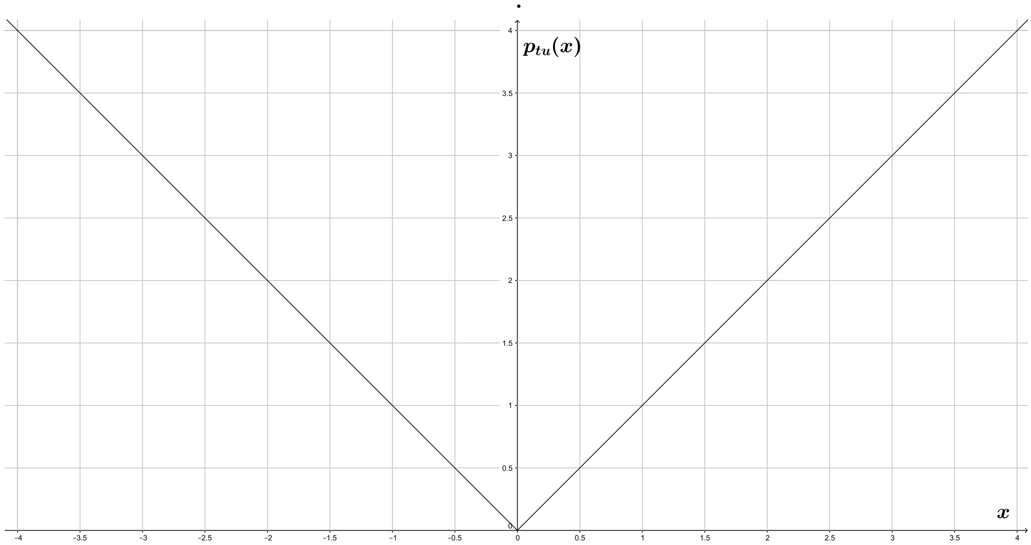

Figure 4.3: Pricing model of a storage, which is denoted by the absolute function

The piece-wise pricing function 4.5 $p_{tu}(x)$ is replaced by a new continuous pricing variable p', which has to adopt the values similarly to the storage pricing model 4.5 [Bis16, Bis09, Wil13]. If and only if $p_{tu}(x)$ is not limited by other constraints, the constraints, denoted by the inequalities 4.6, linearise the pricing model for each contract u.

$$\begin{aligned} p'_{tu} &\geq m_{tu} * x_{tu}, \forall t \leq T \wedge \forall u \in U_{St} \\ p'_{tu} &\geq -n_{tu} * x_{tu}, \forall t \leq T \wedge \forall u \in U_{St} \end{aligned} \tag{4.6}$$

For positive values, p'_{tu} adopts the functional value in the upper inequality. The lower inequality will obtain a negative value which is obsolete because of the upper constraint. By contrast, negative values of x_{tu} set the lower inequality as functional value of $p_{tu}(x)$ and the upper inequality becomes negative and is not regarded by the linear program.
As a last adaptation step, p'_{tu} replaces $p_{tu}(x)$ in the objective function. This leads to a minimization of the constrained value of p'_{tu}.

$$\text{minimize } Z(\mathbf{x}) = \sum_{t=1}^{T} \sum_{u \in U} p'_{tu} \tag{4.7}$$

Since the objective function minimizes all pricing variables p'_{tu} and each decision variable x_{tu} is limited by the bounds, each p'_{tu} will adopt a reasonable value and a feasible solution will be yielded by a linear solver.

As illustrated in figure 4.2, the storage volumes have to be regarded as new constraints because if the minimum volume V_{min} or maximum volume V_{max} is reached, natural gas cannot be withdrawn or injected respectively. Regard that a storage system operator can also constitute a cushion, such that there is an amount of natural gas left in the storage to maintain pressure inside the storage. Thus the following constraint is added:

$$V_{min} \leq V - \sum_{t=1}^{T} x_{tu} \leq V_{max}, \forall u \in U_{St} \tag{4.8}$$

Equation 4.8 limits the decision variables inside the physical volume boundaries while regarding the current amount of gas inside the storage (working gas).
The developed extension faces the problem of to model a piecewise objective function containing an absolute function for negative x_{tu} in order to regard injection for storages. For supply contracts the pricing factor n_{tu} can be set to 0 such that a general pricing variable is obtained.

4.4 Imbalance Penalty

Up to this point it has been assumed that the gas amounts to operate with have to net out K^{Σ} over planning horizon T. In the former model formulation, imbalances are rather disregarded since the tight balancing constraint 3.8 enforces that the sum of all decision variables over the planning horizon T are equal to K^{Σ}.
By assuming this, a problem concerning resolvability of the linear program arises. Suppose the case that a K^{Σ} cannot be balanced by the total sum of all decision variables even if they regard all bounds. As a result, the danger of an infeasible solution is higher because constraint 3.8 may be violated. In reality, it is often hard to satisfy the imbalance completely, too. Therefore the tight constraint needs to be relaxed.
As mentioned in 2 imbalances are penalized by the market area manager. He releases for each day the penalty fees, which have to be paid by the trader if he is not able to balance the network.

Example 4.4.1: Consider that a natural gas trader has to balance an amount of K^{Σ} and he operates with the supply contracts u_1, u_2, u_3. The contractual bounds, however, limit the gas amounts that can be extracted from each contract such that K^{Σ} cannot be reached. The remaining amount is the imbalance, as figure 4.4 shows.

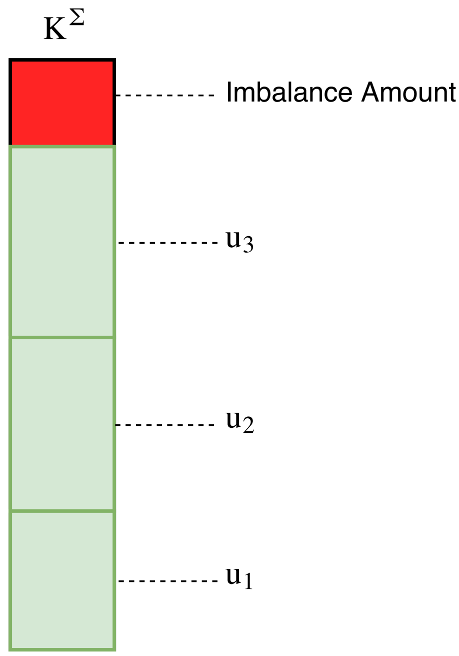

Figure 4.4: The given contracts are not sufficient to balance K^Σ. The resulting imbalance amount needs to be penalized.

The penalty fee is invoiced the day of operation, i.e. usually the fees are not known in the planning phase. To obtain feasible solutions, though, an imbalance penalty is added to the objective.
One approach is the cost calculation of storages, as denoted in the constraints 4.5. On the left-hand side, an auxiliary variable Φ' would denote the total imbalance penalty. On the right-hand side an imbalance cost factor would be multiplied with the imbalance amount. Similarly to the pricing of storages, which must hold for positive and negative decision variables x_{tu}, the following constraints would extend the MIP.

$$\begin{aligned} \Phi' &\geq \iota^+ \left(K^\Sigma - \sum_{t=1}^{T} \sum_{u \in U} x_{tu} \right) \\ \Phi' &\geq -\iota^- \left(K^\Sigma - \sum_{t=1}^{T} \sum_{u \in U} x_{tu} \right) \end{aligned} \tag{4.9}$$

Consider now that many contracts, e.g. 5000 contracts, are passed to the MIP. For both constraints $T \cdot 5000$ decision variables have to be summarized twice. With more and more contracts the computational effort raises because both constraints have to be satisfied.
Another option is a relaxation of constraint 3.8 by adding so called non-negative surplus or slack variables [HL01, Wil13].

The implementation of surplus variables have the advantage that only the balancing constraint, as denoted in equation 3.8 and the objective function needs to be modified. Furthermore, the summary of all decision variables over the planning horizon is done once, which reduces the execution time for large problems. First, the following variables and parameters introduced.
Let

- ϕ^- slack variable to reduce a too high amount to K^Σ, $\phi^- \geq 0$
- ϕ^+ slack variable to raise a missing amount to K^Σ, $\phi^+ \geq 0$
- ι pricing for daily imbalances, $\iota \geq 0$

Both slack variables are added on the left-hand side of constraint 3.8, which results in the following modified constraint:

$$\sum_{t=1}^{T} \sum_{u \in U} x_{tu} + (\phi^+ - \phi^-) = K^\Sigma \tag{4.10}$$

In case that the sum of all decision variables cannot balance the one of the surplus variables compensate the missing amounts. If K^Σ is positive and there is a missing amount ϕ^+ balances. ϕ^- forms the imbalance penalty in case of K^Σ is negative.

Example 4.4.2: Consider that there is a single planning phase where $K^\Sigma = 10$ has to be balanced. The upper bounds of two different contracts limit the decision variables to $x_{11} = 4$ and $x_{12} = 4$. Since the equality constraint 4.10 has to be satisfied, the surplus variable Φ^+ will adopt to $\Phi^+ = 2$.

To penalize imbalances, both surplus variables are added to the objective function and multiplied with a non-negative cost factor ι. Equation 4.11 denotes the adapted objective function and contains the imbalance penalty.

$$\begin{aligned}
&\text{minimize } Z(\mathbf{x}, \mathbf{y}, \mathbf{p'}, \phi^+, \phi^-) = \underbrace{\sum_{t=1}^{T} \sum_{u \in U} p'_{tu}}_{\text{operational costs}} + \underbrace{\gamma(\mathbf{y})}_{\text{take-or-pay penalty}} + \underbrace{\iota\Big(\phi^+ + \phi^-\Big)}_{\text{imbalance penalty}} \\
&\text{subject to} \\
&\qquad 4.2 \quad \text{(contractual limits and deactivation of contracts),} \\
&\qquad 4.6 \quad \text{(pricing for positive and negative decision variables),} \\
&\qquad 4.8 \quad \text{(volume constraint for storages),} \\
&\qquad 4.10 \quad \text{(relaxed balancing constraint),}
\end{aligned} \tag{4.11}$$

The minimization of the objective function reduces the total costs, which means that imbalance penalties are minimized, too. Since the balancing constraint 3.8 is relaxed the solver yields a feasible solution although a set of contracts cannot balance the required amount K^{Σ}. If imbalances occur, then additional costs will be added to the solution.
A similar extension is needed if the natural gas trader needs a phase-wise balancing for all or certain phases of the planning horizon T.

4.5 Phase-wise Imbalance Penalties

It has been assumed that K^{Σ} covers the natural gas amount for the whole planning phase T. Another option a trader can desire is a phase-wise balancing. This probrelem could occur if hourly balancing is more attractive for the trader than a daily balancing or if some contracts can be operated in certain phases [BD16].
Assume for this case that for each phase a source sum k_t^{Σ} is known. Furthermore suppose that if every phase-wise sum k_t^{Σ} is balanced, then the whole day is balanced. Similarly to the imbalance penalty of section 4.4, there are phase-wise penalties by adding surplus variables and phase-wise constraints. The following parameters are added to the mixed integer programming problem.
Let

$\mathbf{k}^{\Sigma}$ Vector of source sums $\mathbf{k}^{\Sigma} = \left(k_1^{\Sigma}, \ldots, k_T^{\Sigma}\right)$

k_t^{Σ} phase-wise source sum which needs to be net out

h_t^- slack variable to reduce a too high amount to k_t^{Σ}, $h_t^- \geq 0$

h_t^+ slack variable to raise a missing amount to k_t^{Σ}, $h_t^+ \geq 0$

For each phase, a relaxed phase-wise balancing constraint is added to the mixed integer program. This sum over the planning horizon T, i.e. $\sum_{t=1}^{T}$, has to be relaxed. As equation 4.12 indicates, these constraints look like the phase-wise balancing constraint.

$$\sum_{u \in U} x_{tu} + (h_t^+ - h_t^-) = k_t^{\Sigma}, \quad \forall t \leq T \tag{4.12}$$

The summary of all h_t^+ and h_t^- is multiplied with a penalty factor ι. Since the real penalty pricing will be released at the next day, the phase-wise penalty factor is equal as the ι of the daily imbalance penalty, which should adopt to a high value. This avoids that imbalances are preferred by the solver. The obtained mixed integer programming problem is denoted as the following:

$$\text{minimize } Z(\mathbf{x},\mathbf{y},\mathbf{p}',\phi^+,\phi^-,\mathbf{h}^+,\mathbf{h}^-) = \underbrace{\sum_{t=1}^{T}\sum_{u\in U} p'_{tu}}_{\text{operational costs}} + \underbrace{\gamma(\mathbf{y})}_{\text{take-or-pay penalty}} + \underbrace{\iota\left(\phi^+ + \phi^-\right)}_{\text{imbalance penalty}} + \underbrace{\iota\sum_{t=1}^{T}\left(h_t^+ + h_t^-\right)}_{\text{phase-wise imbalance penalty}} \tag{4.13}$$

subject to

4.2 (contractual limits and deactivation of contracts),
4.6 (pricing for positive and negative decision variables),
4.8 (volume constraint for storages),
4.10 (relaxed balancing constraint),
4.12 (relaxed phase-wise balancing constraint),

By combining the imbalance adaptation of section 4.4 with the phase-wise imbalance penalty of this section, the solver tends to net out phase-wise imbalances because the penalty costs are higher when there are several penalty misses than a single one (imbalance penalty over all phases).
All cost terms of the objective function have considered operational costs or compensation costs if either contracts are not regarded or if the trader cannot balance the network. A further important aspect is capacity booking in case a contract has defined a physical grid point as location.

4.6 Capacity Costs and Additional Capacity Bookings

Natural gas flows in a network are restricted by physical capacities and need to be booked sufficiently to guarantee the transport of gas. Since the liberalization of the natural gas markets, the trader does not need to know about the whole network [ZS09].

There are two types of capacities: On the one hand, entry capacities need to be booked, for example if the trader imports gas via a physical border point or withdraws gas out of a storage. On the other hand, there are exit-capacities, which are needed in case a trader injects gas into a storage or exports gas via a physical border point. If the trader exceeds those capacities, he has to book further capacities. Especially the trading via the virtual trading point and the trading contracts with consumers are interesting, because the capacities are already known and there are no additional bookings necessary.
Since capacities form a further type of costs, the linear program has to be extended with costs for entry and exit capacity bookings. There are some special features that have to be considered for the mathematical modelling. As mentioned, some contracts regard already the whole capacity and no further bookings are needed, e.g. if the trading location is via the virtual trading point. For the mathematical modelling this has certain consequences.
First assumptions for including capacity costs to the model have to be made. Suppose that the capacities are known for the planning horizon T and that these capacities are valid over the whole planning horizon.
Let

$C(x)$	capacity costs function
c_f	fixed capacity costs
$C_u^{(\epsilon)}$	booked entry capacity for a contract u over planning horizon T, $C_u^{\epsilon} \geq 0$
$C_u^{(\chi)}$	booked exit capacity for a contract u over planning horizon T, $C_u^{\chi} \geq 0$
c_ϵ	additional pricing factor, $c_\epsilon \geq 0$
c_χ	additional pricing factor, $c_\chi \geq 0$

Before the mathematical formulation is modified, it is assumed that if a contract is not coupled with a physical grid point, i.e. there are unlimited capacities, then $C_\epsilon = +\infty$ and $C_\chi = +\infty$. In other words, an additional booking shall be deactivated for grid points like a virtual trading point.
The capacities modification includes the introduction of a new capacity costs function $C(\mathbf{x})$:

$$C(\mathbf{x}) = c_f + \sum_{u \in U} c_\epsilon * max(\sum_{t=1}^{T} x_{tu} - C_u^{(\epsilon)}, 0) - c_\chi * min(\sum_{t=1}^{T} x_{tu} + C_u^{(\chi)}, 0) \qquad (4.14)$$

$C(\mathbf{x})$ computes the total capacity costs and regards exceeded capacities (which generate additional costs). A problem here is that the maximum and minimum

function must be transformed because both functions are non-linear functions.
On beforehand we add $C(\mathbf{x})$ to the linear model. The modified objective function is denoted as in equation 4.15:

$$\text{minimize } Z(\mathbf{x}) = \sum_{t=1}^{T} \sum_{u \in U} p_{tu}(x_{tu}) + C(\mathbf{x}) \tag{4.15}$$

In order to replace the non-linear functions, the maximum and minimum function need to be linearised. This is achieved by replacing each two auxiliary non-negative continuous variables $w_{tu}^{(\epsilon)}$ and $w_{tu}^{(\chi)}$, which replace the non-linear functions [Wil13, Bis16, Bis09]. Both auxiliary variables need to adopt the values of each non-linear function. Therefore the following constraints are added:

$$\begin{aligned} w_u^{(\epsilon)} &\geq c_\epsilon * \sum_{t=1}^{T} \left(x_{tu} - C_u^{(\epsilon)}\right), \forall u \in U \\ w_u^{(\chi)} &\geq -c_\chi * \sum_{t=1}^{T} \left(x_{tu} + C_u^{(\chi)}\right), \forall u \in U \\ w_u^{(\epsilon)}, w_u^{(\chi)} &\geq 0 \end{aligned} \tag{4.16}$$

The added constraints work similarly to the absolute value transformation in the storage pricing functions in section 4.3. First of all if the decision variables in between the contractual amounts, the non-negativity constraint will be active because both constraints are negative. If, however, one capacity is exceeded, one constraint will adopt the correct value, while the non-negativity constraint becomes active for the other constraint.
In case that there are no capacities, either the pricing factors c_ϵ and c_χ can be set to 0 such that the first two constraints become non-negativity constraints. The constraints are redundant, then. Another option is to set the booked capacity amounts to ∞ (or a sufficient huge positive value), which results to negative right-hand sides for the first two constraints. The non-negativity constraints will force the auxiliary to adopt non-negative values.
Replacing the non-linear functions with the auxiliary variables and adding them to the capacity cost function is denoted as in equation 4.17.

$$C(\mathbf{x}) = c_f + \sum_{u \in U} w_u^{(\epsilon)} + w_u^{(\chi)} \tag{4.17}$$

Since the objective function tries to minimize the costs, the algorithm will try to keep the amount of capacity bookings as low as possible which keeps additional costs low.

4.7 Market Options

Besides operating with supply and demand contracts a trader could also purchase natural gas via a natural gas market, e.g. a spot market. This case occurs if the prices are attractive for the trader at gas market. Two considerations have to be made: First of all, a trader may purchase or sell gas, as much as he desires. This means that it is possible that there are no bounds. Secondly, a trader decides if a contract is attractive by setting a pricing threshold. If the threshold is violated, a contract becomes unattractive. For the implementation of market options, it is assumed, that market options act quite similar as natural gas contracts and the purchase case is considered in this subsection. Let

U_m set of market buy options, $U_m \subseteq U$

ξ_u pricing threshold, which can be set by the trader $\xi_u \geq 0$

m_{tu} pricing factor, as known from the pricing model function, $m_{tu} \geq 0$

ϵ very small positive real value, $\epsilon \geq 0$

The control parameter ξ_{tu}, set on beforehand, determines the upper bound for an attractive pricing. So if $m_{tu} < \xi_{tu}$, the price is attractive and the market option shall be active. If the threshold is breached, the market option shall not be used. Since a strict inequality cannot be inserted into a constraint, ϵ is added. This replaces the strict inequalities either to $\leq$ or $\geq$.
Assume that in case of an attractive market price, the trader purchases gas via a spot market rather than operating with a contract. In order to determine the interval, the bounds can be set as desired, e.g. $L_u = 0, U_u = M$ for purchasing gas, where M may be a large value. The trader can define the maximum amount of gas he wants to take from the spot market.
Consider now that a market price may be unattractive at a certain day. If so, the linear program shall ignore the market. This can be achieved by adding the following constraints:

$$x_{tu}\left(m_{tu} - \xi_u + \epsilon\right) \leq 0, \quad u \in U_m \wedge \forall t \leq T \tag{4.18}$$

Remember that the binary variable, which is introduced in section 4.1, is used to disable contracts by the linear programming algorithm. This binary variables needs also to be considered for market options. Suppose the case if $y_u = 1$. If $\xi_u < m_{tu}$, then constraint 4.18 will be satisfied, because the term in the parenthesis results in a negative value.

If, however, the market prices are unattractive $\xi_u \geq m_{tu}$, then the whole term inside the parenthesis becomes positive which results in a violation of constraint. As a consequence unattractive market options are cut off the solution space.
The mentioned cases considered only market options, so far. In case the constraint shall be deactivated for natural gas contracts, though, the parameters have to be set properly, so that the constraint will not cut off solutions. If u is a contract, then $\xi_u = M$, where M is a sufficient large positive value. This will enforce that the left-hand side is always negative and the constraint is satisfied.

4.8 An aggregated Mixed-Integer Programming Problem

The basic linear programming problem of section 3.3 has assumed that there is a total balance and contractual bound constraint, as denoted in equation 3.8 and equation 3.9, respectively. To cover different contract types, penalties and capacities, the model has been adapted step-by-step in the previous subsections
For simplicity, all proposed extensions have been considered isolated. In this section, these extensions are concluded to one linear programming problem. Furthermore this section describes the data setting and proposes a draft for the implementation. Modifying linear programming problem of section 3.3 by the extensions has the effect that several variables has to be added to the definition. For each contract, a binary variable is added, which determines if a contract is used or not. Together with these binary variables, auxiliary variables complement the linear programming problem, which are set by the constraints, e.g. the pricing variable p'_{tu}. Equation 4.19 describes the modified linear programming problem.

$$\text{minimize } Z(\mathbf{x}, \mathbf{y}, \mathbf{p'}, \mathbf{h}^+, \mathbf{h}^-, \phi^+, \phi^-, \mathbf{w}^\epsilon, \mathbf{w}^\chi) = \underbrace{\sum_{t=1}^{T} \sum_{u \in U} p'_{tu}}_{\text{operational costs}} + \underbrace{\gamma(\mathbf{y})}_{\text{take-or-pay penalty}} + \underbrace{\sum_{t=1}^{T} \left(h_t^+ + h_t^-\right)}_{\text{hourly imbalance penalty}} + \underbrace{\iota\left(\phi^+ + \phi^-\right)}_{\text{imbalance penalty}} + \underbrace{C(\mathbf{x})}_{\text{additional capacities}} \tag{4.19}$$

subject to

4.2 (contractual limits and deactivation of contracts),
4.6 (pricing for positive and negative decision variables),
4.8 (volume constraint for storages),
4.10 (relaxed balancing constraint),
4.12 (relaxed phase-wise balancing constraint),
4.16 (additional capacity booking)

Suppose there is a pre-processing unit, the core solver containing the linear optimization definition and a post-processing unit. While a linear solver controls all variables such that an optimal solution is found, the fixed parameters have to be set and passed to a solver before the algorithm. The following parameter are contractual data parameters:

m_{tu} Operational costs for injection into a storage and supply contracts,

n_{tu} operational costs for withdrawal out of a storage,

L_u contractual daily lower bound,

U_u contractual daily upper bound,

V current volume of a storage,

V_{min} minimum volume of a storage and

V_{max} maximum volume of a storage

Depending on the contract, the listed parameters are set by a default value, e.g. the volumes for supply contracts, or with the specific values. In case of long- and mid-term contracts, one could think to set the parameters automatically in a pre-processing step, since they are not volatile. However, consider that if there is a contract, which is only valid for one day. The trader is then forced to add these parameter manually for each day. A similar problem occurs for capacities and the market pricing threshold. If those are only known for the next day, they will have to be set manually. This includes the following parameters:

c_f fixed capacity costs

C_u^{ξ} booked exit capacity amount

c_u^{ξ} pricing for additional exit capacities

C_u^{ξ} booked entry capacity amount

c_{ξ} pricing for additional exit capacities

ξ market pricing threshold

This means for the linear program, that pre-processing steps are needed. Furthermore the solution of the linear program needs to be translated, such that the results can be observed by a natural gas trader. Figure 4.5 sketches a process flow of a simple implementation.

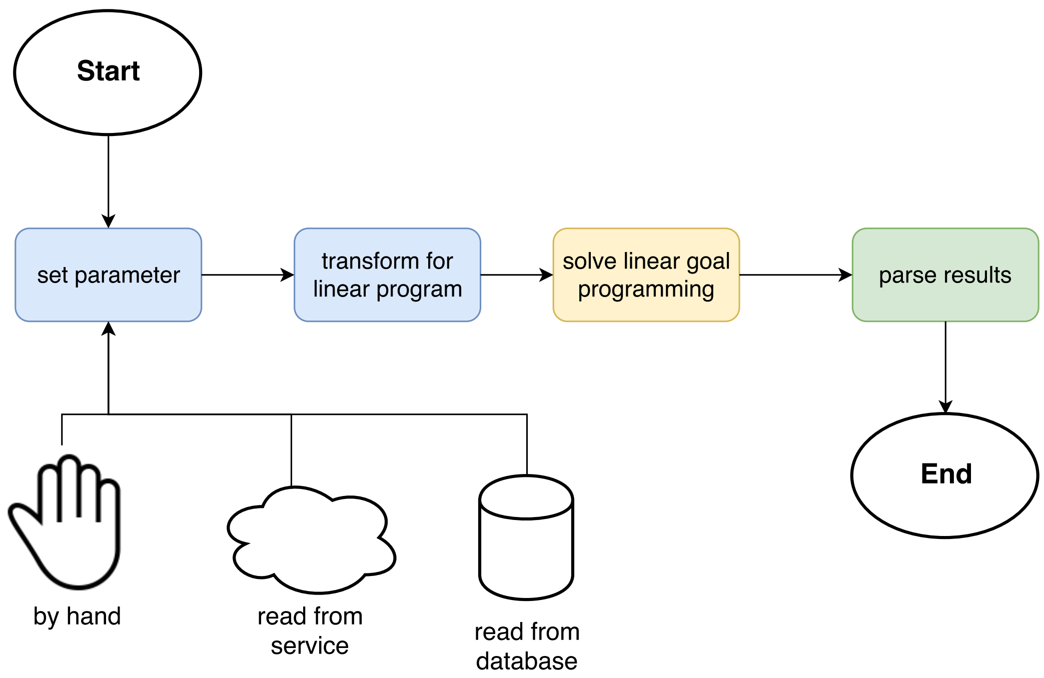

Figure 4.5: Flow diagram for the linear program.

If there are no default data values, this has to be done manually or automatically via a remote service, which offers the data or an own database. These values may need to be transformed into a format, which is readable by the underlying solver.

This transformation concerns mainly the setting of the parameters in the constraints and objective function. These two steps conclude the pre-processing phases. This transformation is passed to the solver, which can be executed afterwards. The linear solver yields either a feasible or infeasible solution, which is parsed for the output, for instance a graphical user interface.
One extension has been spared out in this section: The choice of strategic bounds, which is discussed in section 4.9. One solution is to set these bounds manually for each planning horizon T. Suppose that a natural gas trader could operate with hundred or thousand contracts per month. For each day, these strategic bounds need to be set manually, which can be an expensive task if the bounds for many contracts have to be set.

4.9 Strategic Bounds

Up to here, only a contractual bound has been regarded in the former mathematical models. Since it is a contractual agreement, this bound does not change in the planning phase. In addition, it has been assumed that there were no capacity costs. Since the physical capacities are limited a trader has to book capacities for the transportation or withdrawal or injection of natural gas in a certain phase.
It may occur that a trader wants to intervene and set the boundaries manually in order to curtail or extend the amount of taken gas. Consider, that a trader wants to operate only with a certain amount of natural gas or he wants to operate with a certain contract but the solution of the linear program does not include it. Then he can curtail other contracts to regard his wanted contracts. In case of an extended amount of gas, the trader needs to book further capacities because the contractual bounds cover only the booked capacities in a given interval. In this subsection, the linear program is extended by capacity bookings and strategical bounds.
The model needs therefore new parameters. Let

$L_u^{(c)}$ contractual lower bound for contract u over planning horizon T (former L_u)

$U_u^{(c)}$ contractual upper bound for contract u over planning horizon T (former U_u)

$L_u^{(s)}$ strategic lower bound for contract u over planning horizon T

$U_u^{(s)}$ strategic upper bound for contract u over planning horizon T

Combined with the contractual bounds a trader has the option to change the bounds by defining strategic bounds. Along with the assumptions four cases that determine the contractual gas amount are identified which are presented in figure 4.6.

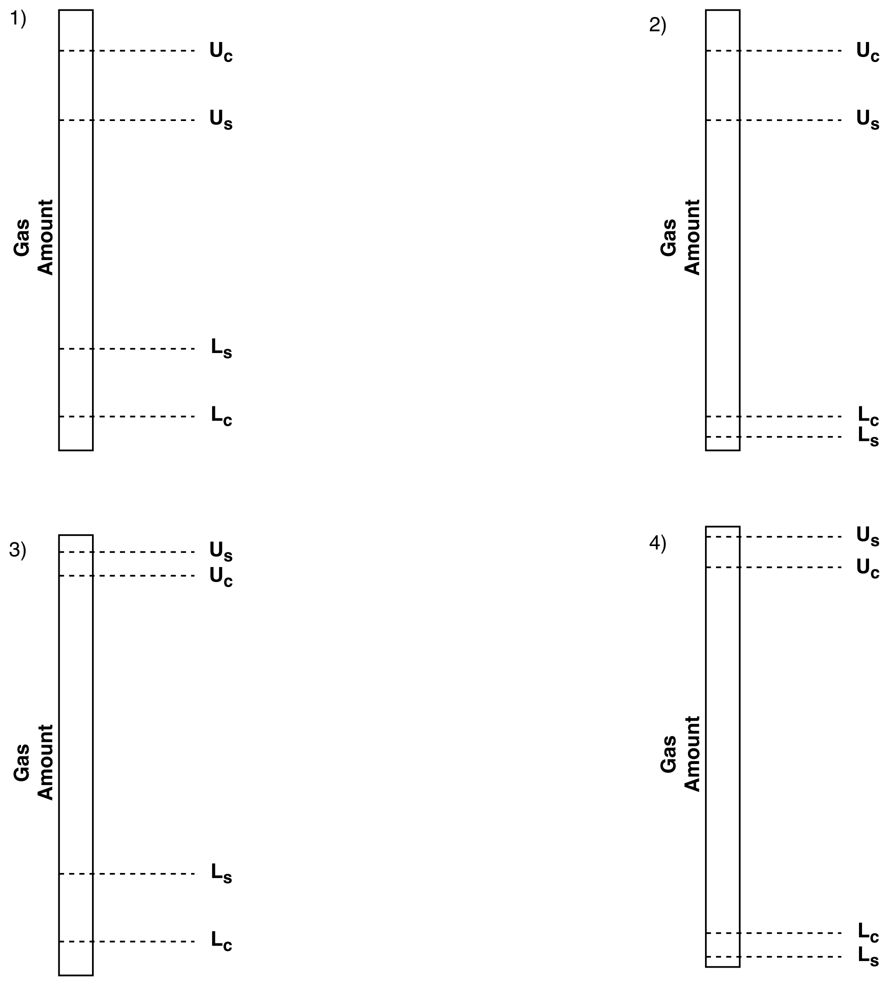

Figure 4.6: Different cases of setting strategic bounds. In this figure, the contractual bounds are always fixed, while the strategic bounds are flexible.

Case 1 illustrates a curtailment of the contractual amount from both sides. Both, the strategic lower and upper bound, reduce operable amount. Case 2 presents a reduction of the upper bound and allows to select less gas from the contract. Case 3 shows an increase of the lower and upper bound by the strategic bounds. Case 4 illustrates widens the margin of operable gas, while the contractual bounds shall be disregarded.
To select the corresponding bound, the bounds can be determined in a preprocessing step by substituting manually the contractual bounds by the strategic bounds. This chapter has proposed conceptual extensions that should cover the complex day ahead balancing problem. The simplified application of section 3.3 has to be merged with this modifications. Take-or-pay rates, storage contracts, imbalance penalties and capacity costs have been gradually added to the linear optimization problem. As a result, a mixed-integer programming problem has been developed. The applicability is tested in the following chapter 5.

Chapter 5

Implementation and Numerical Results

While the preceding sections covered the conceptual work of finding a cost-optimal contract selection, it has to be tested if the proposed mixed-integer programming problem yields reliable results. Furthermore it is also informative in how far the amount of variables and constraints affect the execution time of the proposed optimization problem.
One way to check if the linear program works is to pass some data to the linear program. In this section a prototypical concept is proposed. First, the prototype design and the used tools and hardware are presented. In addition, the used data and the data structure is discussed. Finally, a performance test shall reveal the efficiency of the implementation.

5.1 Prototype Design

Figure 5.1 sketches the design of an object-oriented implementation. The main component is an abstract class called **AbstractSolver**, which hides the underlying solver. Any concrete class can contain and use another solver, for example, the **MinimizingCostsSolver** solver uses GLPK There are further solvers, e.g. IBM CPLEX or the Gurobi Solver library which can be exchanged. The polymorphic structure allows to exchange and implement solvers for different problems.
Before a solution can be obtained, the parameter, variables, constraints and finally the objective function have to be initialized. Most solvers support a mathematical modelling language, for example the *Advanced Interactive Multidimensional Modelling*

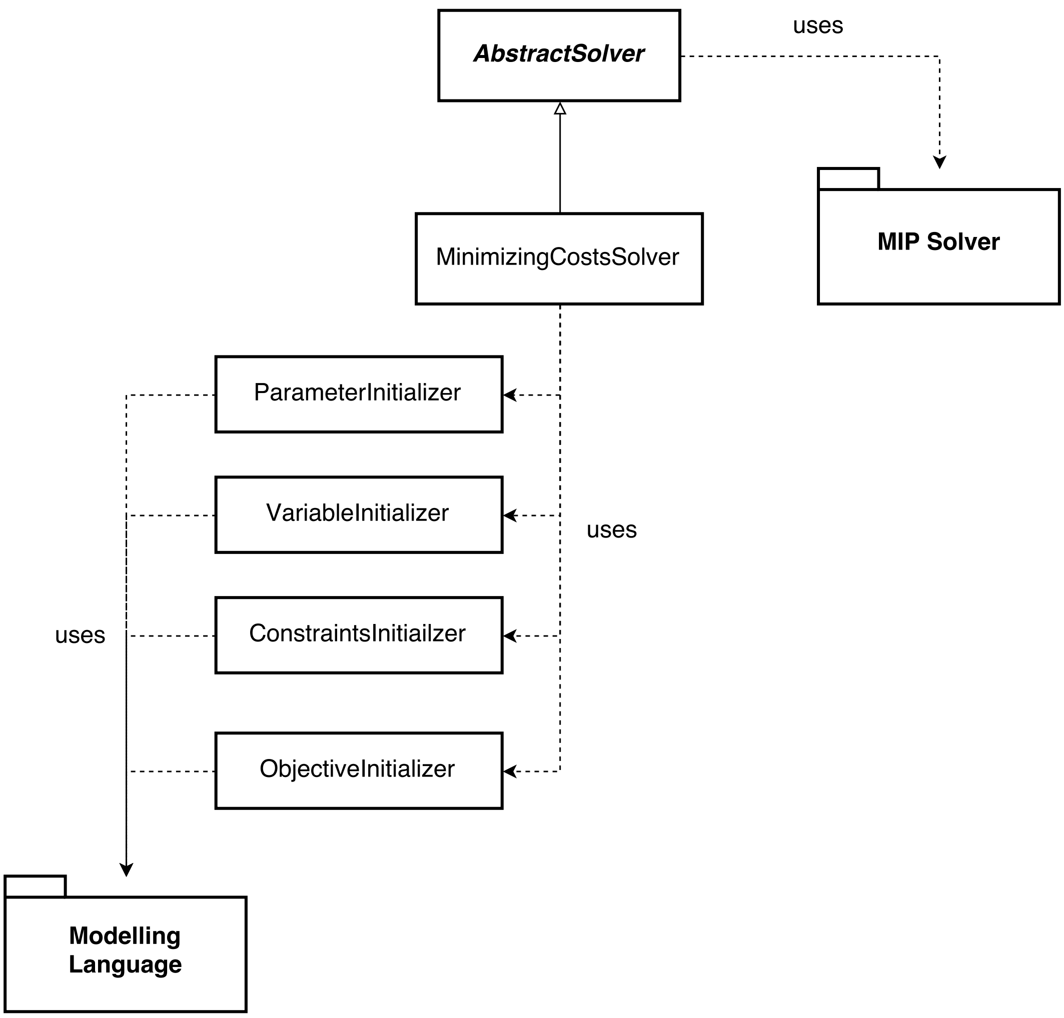

Figure 5.1: UML draft for the prototype implementation

System (AIMMS) or *A Mathematical Programming Language* (AMPL). These modelling languages unify the formulation of optimization problems. To formulate the components, i.e. defining required parameters, variables, constraints and the objective function of section 4.8, initializers are passed to the corresponding solver. For each optimization problem a specialized initializer is available such that the solver class is adaptable. These initializer classes contain the corresponding parts of the model and when the solution process is executed, the formulation is passed to the solver.

5.2 Implementation Tools

The prototype is implemented in Python and uses Pyomo a python-based optimization modelling language [HWW11]. In contrast to AIMMS and AMPL, which

are algebraic modelling languages, the optimization problem can be implemented in Python, which allows to use the features and further libraries of the high-level programming language.
Pyomo is independent from the underlying solver. The library transforms parameters, variables, constraints and the objective function in the appropriate solver format. This means that the contractual data, which is needed to solve the day-ahead balancing problem, can be pre-processed, as illustrated in figure 4.5, and passed to the used solver.

5.3 Data

In order to test and to experiment with the MIP, datasets have to be created. These datasets are used to parametrise the optimization problem, for example with the lower and upper bounds or the pricing coefficients. Some data is required to solve the MIP, e.g. K^{Σ}, T and the contracts. Table 5.1 lists the data specification which are passed to the prototype.

Parameter	**Type**	**Required/Optional**
Daily Balance Amount K^{Σ}	Number	required
Planning Horizon T	Number	required
Contracts	Set/List of Contract Objects	required
Imbalance Penalty ι	Number	optional
Hourly Balance Amount K_t^{Σ}	Set of Numbers	optional
Fix Capacity Costs c_f	Number	optional

Table 5.1: Parameter Setting for the prototype

Each contract of the list or set has to define a name and the pricing. Depending on the contract type the supply pricing m_{tu} (supply, market buy options and storage contracts) or the demand pricing n_{tu} (demand and storage Contracts) must be parametrised. In case they are not required, default values will be set. For storage contracts the volume bounds and current volume have to be regarded. Parameters like capacities are optional.
Since the pricing values should be realistic, the pricing coefficients and capacities are derived from a natural gas market exchange platform and a capacity booking platform, respectively. As a result, a simulation can be executed with the proposed

model to check if a reliable optimal solution is obtained.

5.4 Simulated Scenario

The basic application of section 3.3 has given a first insight of a model formulation, which yielded a trustworthy solution. A question that occurs is, if the prototype implementation is still able to solve a problem correctly. As discussed in the previous section, the given data, for example contractual bounds, capacities, pricing factors, is simulated. By passing the simulation data to the prototype, the solution itself and the effects of changing parameters are analysed.
It is assumed that the planning horizon is a whole day such that $T = 24$. Let $K^{\Sigma} = 17000$ the amount that has to be net out the next day. Furthermore there are phasing-wise balancing amounts, which are listed in table 5.2. Some phases are missing on purpose, namely phase 1 and phase 21 to 24, because the solver shall try to fill these phases until it reaches the daily balance. In addition, the imbalance penalty factor $\iota = 50$, such that imbalances should be avoided.

Phase	Phase-Wise Balancing Amount
2	794
3	726
4	822
5	632
6	693
7	655
8	647
9	655
10	679
11	784
12	705
13	750
14	702
15	779
16	769
17	757
18	755
19	744
20	739

Table 5.2: Phase-Wise Balancing amounts

The natural gas trader have 10 contracts available to find a cost-optimal distribution which balances the network. Table 5.3 presents the given supply contracts. One option is to buy natural gas via the exchange market by selecting the market buy option, where $\xi = 9$ denotes the pricing threshold. Both storages have capacities above or below the contractual bounds, which means, that additional capacities shall be bookable. The supply contracts Flex 2, Flex 3 and Fix 1 have no capacities, since a VTP is the contractual grid point of execution. The remaining contracts are imported via a border grid point. Therefore the trader has allocated capacities for those contracts. Last but not least, the supply pricing factors m_{tu} are moving in a range between 8 and 16 cost units.

Contract	**Lower Bound**	**Upper Bound**	**Further Parameters**
Market Buy Option	0	5000	$\xi = 9$
Storage 1	-500	500	$V = 400$ $V_{min} = 0$ $V_{max} = 800$ $C_u^{\epsilon} = 500$ $C_u^{\chi} = 500$ $c_{\epsilon} = 0.85$ $c_{\chi} = 0.85$
Storage 2	-800	800	$V = 1000$ $V_{min} = 0$ $V_{max} = 1000$ $C_u^{\epsilon} = 400$ $C_u^{\chi} = 500$ $c_{\epsilon} = 0.85$ $c_{\chi} = 0.85$
Supply Flex 1	10	2120	$\theta_u = 1.0$ $C_u^{\epsilon} = 1800$ $c_{\epsilon} = 1.20$
Supply Flex 2	20	2500	$\theta_u = 1.0$
Supply Flex 3	20	1960	$\theta_u = 1.0$
Supply Flex 4	10	2290	$\theta_u = 1.0$ $C_u^{\epsilon} = 1000$ $c_{\epsilon} = 1.20$
Supply Flex 5	0	2500	$\theta_u = 1.0$ $C_u^{\epsilon} = 2000$ $c_{\epsilon} = 1.92$
Supply Fix 1	2010	2010	$\theta_u = 1.0$ $C_u^{\epsilon} = 2010$ $c_{\epsilon} = 0.95$
Supply Fix 2	2290	2290	$\theta_u = 1.0$

Table 5.3: Simulated contractual reference data

The prototype implementation yields an optimal solution with $Z = 150275.37$ cost units. Table 5.4 lists the optimal contract setting.

Phase	Contract	Amount	Price Factor	Sum
1	Market Buy Option	2451	8.33	20 416.83
2	Supply Flex 4	794	8.41	6677.54
3	Supply Flex 1	102	8.56	873.12
	Supply Flex 5	624	8.21	5123.04
4	Supply Flex 1	822	8.76	7200.72
5	Supply Flex 5	632	8.44	5334.08
6	Supply Flex 2	693	8.97	6216.21
7	Supply Fix 1	655	10.12	6628.60
8	Supply Flex 3	647	8.33	5389.51
9	Supply Fix 2	655	8.37	5482.35
10	Supply Flex 1	679	8.14	5527.06
11	Supply Flex 4	206	9.86	2031.16
	Storage 1	578	10.02	5791.56
12	Supply Fix 2	705	8.89	6267.45
13	Supply Fix 1	750	9.01	6757.50
14	Supply Fix 2	151	9.67	1460.17
	Supply Flex 3	551	9.92	5465.92
15	Supply Fix 2	779	8.16	6356.64
16	Supply Fix 1	605	9.41	5693.05
	Supply Flex 2	164	8.82	1446.48
17	Supply Flex 2	757	9.32	7055.24
18	Supply Flex 2	755	9.31	7029.05

19	Supply Flex 5	744	8.24	6130.56
20	Storage 1	400	10.11	4044.00
	Storage 2	11	10.28	113.08
	Supply Flex 1	197	10.13	1995.61
	Supply Flex 2	131	10.89	1426.59
22	Supply Flex 3	762	8.05	6134.10

Table 5.4: Phase-Wise Balancing amounts

Furthermore the solver invoices the following entry capacities for both storages:

Phase	**Contract**	**Additional Capacity**	**Price Factor**	**Sum**
1	Storage 1	50	0.85	47.50
2	Storage 2	189	0.85	160.65

Table 5.5: Additional entry-capacity amounts of the storages

Most of the pricing factors are below 10 cost units which is close to the lower bound of the pricing range. All contracts were regarded because, firstly, those having a contractual take-or-pay-rate would raise the costs and, secondly, the daily balancing amount would be missed. Since the underlying GLPK solver prints the solution status *optimal* and K^{Σ} is reached, it is assumed that the solution is a reliable result.
If the pricing factor of Market Buy Option 1 in phase 1 is replaced by $m_{tu} = 9.33$, the market option becomes unattractive in this phase, because constraint 4.18. Since other prices are unattractive as well, the option is completely disregarded. The remaining amounts of the supply and storage contracts will, which can occur in additional capacity bookings.
Removing the phase-wise balancing amounts of the test dataset causes the search for the cheapest pricing factor independently of the phase. In other words, the phase-wise balancing amounts are prioritised since the penalty costs would raise up the total costs.
This simulation gives a first insight if the extend model is applicable for the given problem. It is possible to yield a cost-optimal solution for the day-ahead balancing

problem. It has to be regarded that this test has used synthetic data. A better way would be to compare the solution to historic decisions and let an expert review the obtained solution. Another aspect is the efficiency of the formulated model, which is tested in the next section.

5.5 Runtime Performance Test

Along with the provided optimal solution, it is desirable to know how fast a solver can provide a solution for the trader. Consider that a natural gas trader passes 500 or 1000 contracts to the solver. If the solver needs too much time, e.g. an hour, to solve the problem then a re-formulation of the mathematical model can help to improve the runtime. First the proposed MIP is measured. Afterwards it is compared with a more complex version of the MIP.
To test the complexity, the amount of contracts is raised, which means that the amount of variables and constraints increases, too. Many solvers provide information concerning the used algorithm or statistics. An important item is the execution time which is observed for this test.
The test is executed on a machine containing an Intel i5-2450M dual core processor. Each core has a clock frequency of 2.5 gigahertz and have access at 8 gigabytes memory. On this machine runs a 64-bit Linux operating system. As mentioned, the prototype implementation uses GLPK to solve the mixed inter programming problem.
For each time measurement, there is a predefined set of 10 to 5000 contracts. Each test set runs 10 times and the average over all single elapsed times is calculated. Table 5.6 shows the results of the performance test for the given MIP.

No. of Contracts	Variables	Constraints	Average Time in seconds
10	571	1056	0.021
25	1351	2601	0.046
50	2651	5176	0.1178
100	5251	10326	0.3656
250	13051	25676	1.98
500	26051	51526	7.165
1000	52051	103,026	29.285
2500	130051	257526	200.864
5000	260051	515026	869.189

Table 5.6: Execution times

The performance results show that the solver is able to yield an optimal solution efficiently. For 1000 contracts the solver needs approximately 30 seconds to find an optimal setting of 50,000 variables and to satisfy over 103,000 constraints. Hence, an amount up to 1000 seems to be appropriate for the day-ahead or intra-day planning. To show the effects of a refinement, suppose that daily and phase-wise imbalance constraints are replaced by the exhaustive imbalance constraints. The modified model is called *MIP with exhaustive constraints*. By passing the same test set to the prototype, the performance result, as denoted in table 5.7 is yielded.

No. of Contracts	Variables	Constraints	Average Time in seconds
10	546	601	0.0236
25	1326	1426	0.078 34
50	2626	2801	0.1326
100	5226	5551	0.4058
250	13026	13801	2.6692
500	26026	27551	8.8864
1000	52026	55051	36.375
2500	130026	137551	242.0586
5000	260026	275051	1073.114

Table 5.7: Execution times with exhaustive balancing constraints

Figure 5.2 illustrates the elapsed average time which are listed in table 5.6 and table 5.7. While for small problems the average runtimes are nearly equal, the complexity rises for large problems. Even though the proposed MIP has nearly twice as much constraints to satisfy, the proposed MIP is faster. The reason for this is that the imbalance variables need to adopt to a value which is at least as high as all decision variables. Hence, the MIP rather seems sensitive to the formulation of the variables and constraints than the amount of constraints.

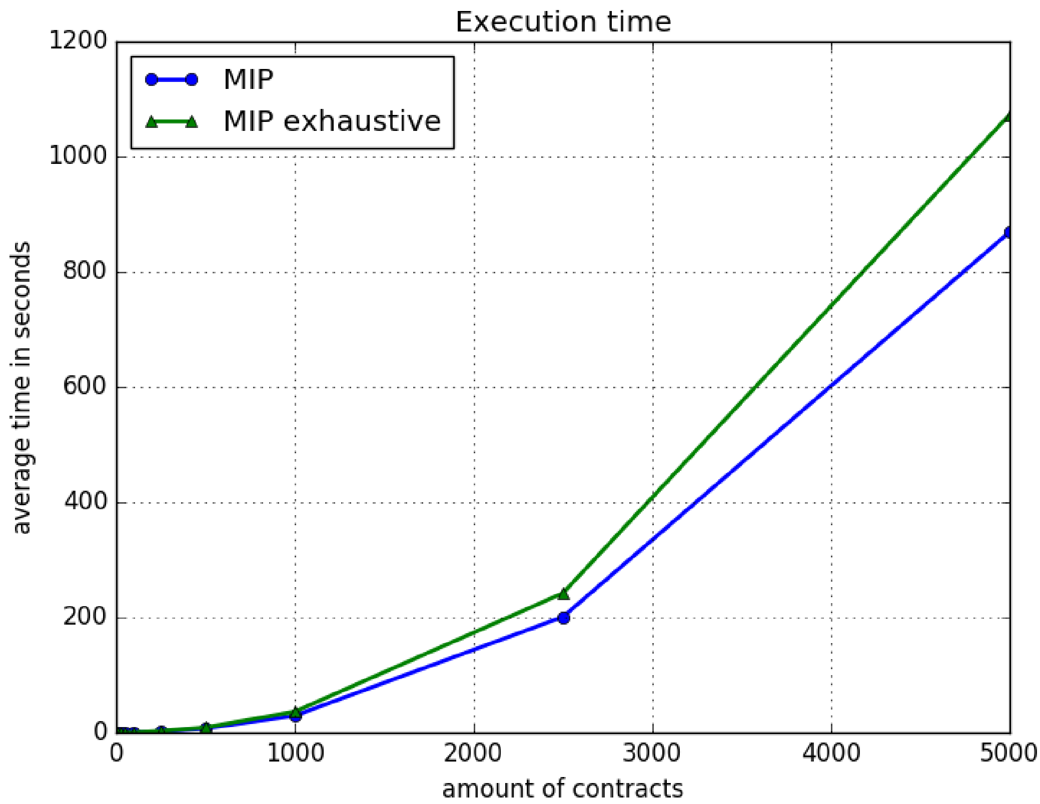

Figure 5.2: Comparison of the proposed MIP and one with exhaustive constraints.

This section covered the prototype implementation of the mixed integer program. This implementation transfers the contractual reference data and parameters into the constraints and solves efficiently the day-ahead balancing problem. The experimental data has yielded reliable results. This results have to be proved with reference data of natural gas traders, though. In the following section 6 the conceptual and implemented work is concluded and discussed.

Chapter 6

Discussion

Based on the problem domain analysis in chapter 2, and the scenario 3.1, a basic linear programming problem has been derived. This model has been gradually adapted while the prototype implementation has shown that the MIP is able to find a cost-optimal contract distribution. In this section, the proposed algorithm and the results are reviewed and discussed. Advantages and disadvantages are debated and unresolved issues are examined for future works.

The first draft has formed a cornerstone of a linear programming programming problem. It should have given a first insight on the problem. Similarly, the literature proposes to start with a basic mathematical model, which is easy to solve. After that the model is extended step-wise [HL01, Ehr05]. This helped to make simple assumptions and test the linear programming with an example.

Since the first draft reveals limitations and day-ahead balancing problem have further considerations, for example storages, dependencies between the decisions or physical capacities, the basic application is extended individually.

The complete cost-optimal day-ahead balancing problem can be partitioned to small ones and each part focuses on an own problem, which can be added to the model and validated afterwards. This methodology simplifies the formulation of a mathematical model which covers the features of the cost-optimal balancing problem. Table 6.1 lists the functionalities that are covered by the proposed MIP.

Task	Covered	Fut. Work
Simple Example	x	
Take-or-pay	x	
Fix Contracts	x	
Storages	x	
Hourly Balancing	x	
Capacities	x	
Imbalance Penalty	x	
Market Options	x	
Multiple Market Areas		x
Strategic Decisions (Best Bounds)		x

Table 6.1: Covered tasks and future tasks of the MIP.

Similarly to the step-wise modelling process, the prototype has been gradually extended. One advantage of this implementation steps is that all adaptations could be integrated with a low effort. Furthermore the model can be tested and compared to previous steps and if the formulation is erroneous, the model can be refined until reliable results are obtained.
In addition, the efficiency of the linear programming problem has been debated. For small problems, such as 500 contracts, a solution is yielded in a few seconds. However, if the problem becomes very large, e.g. more than 5000 contracts, the execution time rises, since more and more variables are generated which need to be computed by the constraints. For large datasets one can think about to refine the current formulation. It is not clear if GLPK uses parallel algorithms. Therefore one could try to execute a parallel solver.
For testing reasons, the mixed-integer program is parametrised with synthetic data. The contractual bounds adopt to random values. To come closely to a realistic execution, randomized prices has been generated whose range have been extracted from the gas exchange market.
Although passing experimental data has shown that the solution is optimal and computed efficiently, the MIP has to be tested with realistic datasets of natural gas traders. In addition an expert should review the decisions of the solver or the solutions need to be compared to target values. This would provide a better estimation, if the proposed algorithm is close to reality.

As table 6.1 outlines, there are further extensions which need to be considered. One aspect are the a cost-optimal distribution of contracts which spans over multiple market areas, which reveals further problems:

- How should the market areas be balanced if there is an imbalance in a single or multiple market areas?
- How should the transfer of natural gas from one market area to another look like?
- Are there any market areas that should be prioritized with respect to have the lowest costs?

One possible approach is to execute the proposed linear program followed by a node-based cost-reducing optimization problem [BB06, Mid07, MMNSD98], which controls could control the natural gas transfers from one market area to another. First there is an isolated optimization of each market area itself. Occurring imbalances could be net out afterwards with the node-based mathematical problem, such that the costs are minimal and the overall network is balanced.
A further open point is the choice of the strategic bounds for contracts. In section 4.9, it is proposed to substitute these bounds manually. If there are multiple contracts, this would be an exhaustive task for a trader, who desires a good strategic bound for daily planning. Depending on the strategy the trader could curtail, expand the bounds or set the strategic bounds equal to the contractual bounds.
Besides the cost-optimal minimizing goal, these bounds could also coupled to an amount oriented goal, e.g. inject at least 50000 volume units into a storage because the buying price for the next month seems to be attractive.
An approach could be the definition of a linear goal programming problem[HL01], which is solved before the proposed mixed-integer programming problem. Compared to the latter, multiple goals are defined, which are part of the objective. This objective is tried to minimize by reaching the goals. If one goal is missed, a penalization factor will raise the *costs* [HL01].
First, the amount oriented goal receives an expected valuation. This valuation is an indicator for the attractiveness for the amount oriented goal, so that the options are compared. In other words, is it better to operate with the contractual bounds or the strategic bounds? In the first step the strategic bound is set equal to the contractual bound. If the daily expected valuation is missed, the strategic bound could be adapted towards the expected value.

The methodology of adapting the strategic bounds is an open problem and a termination condition should be defined, as well. If this gap is closed, the trader needs to determine less parameter and can execute a full-automated system which supports the decision tasks of a natural gas trader.

Chapter 7

Conclusion and Future Work

Determining natural gas amounts in a cost-minimal sense can be one option for a natural gas trader to operate with a given portfolio. In this work a proposal has been presented to find a cost-minimal contract distribution by using linear programming and mixed-integer linear programming.
Before the optimization problem has been approached, the problem domain of the day-ahead balancing problem has been analysed. This has been a necessary step to make assumptions and to identify important parameters and constraints for the cost-optimal contract distribution.
After clarifying the foundations of linear programming and mixed-integer programming, a simplified linear programming problem has been formulated which contains simplified assumptions and constraints.
This simplified version has been modified with extensions which are derived from the problem domain such that we come closer to a realistic scenario. An issue that accompanies with the formulation of the extensions is the fact to keep the optimization problem linear, such that the mathematical program is efficient to solve. So called yes-or-no decisions have led to a mixed-integer programming problem because binary variables have been added to the model. Non-linear functions, e.g. the absolute valuation in the case of storages or minimum and maximum functions, have been transformed into linear constraints such that the optimization problem is still linear.

Besides the open aspects in chapter 6, there are open conceptual works, questions and issues with regard to the linear programming approach. The proposed model has observed a planning horizon for the day-ahead planning.
This has the effect, that a natural gas trader could create plans for a week or a month or even longer. Accompanied to this examination it has to be proved if the strategic goals are compatible with the solutions of the linear program.
The implementation of an extensible tool, that pre-processes the given data, solves the linear optimization problem and post process the results is another future step. In the prototypical implementation most of the data has been set manually. For computational calculations, however, an automatic tooling that fetches the data, pre-processes and passes it to the proposed linear program and post-process it would support the planning process of the natural gas trader.
A further extension could be the integration of the source side of the commodity balance. In this work it has been supposed that the natural gas trader sets the known sources, which obtain a required amount of natural gas, e.g. K^{Σ}. Another extension is reduction of costs on the source side, as well. This can lead to further cost efficiency.
Last but not least, it has been assumed that the given data is deterministic and does not change. The natural gas market is rather volatile, though. In other words, the market situation and the prices are uncertain and these can change during the planning horizon. Thus, the trader may want to see different planning options, such that he is able to react on several planning options in future.

Bibliography

[BB06] Gido A.J.F. Brouns and Alexander F. Boogert. Gas portfolio and transport optimization. Birkbeck working papers in economics and finance, Birkbeck, Department of Economics, Mathematics & Statistics, 2006.

[BD16] Andreas Bräsigk and Martin Drescher. Interviews about the problem domain at ECG Erdgas Consult GmbH, May 2016.

[Bec16] Johannes Becker. Inteview about natural gas pricing models at Exxeta AG, July 2016.

[Bis09] Johannes Bisshop. AIMMS Modeling Guide - Integer Programming Tricks, 2009. Last Access: September 5th, 2016.

[Bis16] Johannes Bisshop. AIMMS Optimization Modeling, 2016. Last Access: September 5th, 2016.

[Cha16] Vivek Chandra. *Gas Pricing & Contracts*, 2016 (accessed June 13th, 2016).

[Com13] EFET Gas Committee. Guide on the features of a successful virtual trading point, 2013.

[Dan63] G.B. Dantzig. *Linear Programming and Extensions*. Princeton landmarks in mathematics and physics. Princeton University Press, 1963.

[Ehr05] M. Ehrgott. *Multicriteria Optimization*. Springer Berlin Heidelberg New York, 2nd edition edition, 2005.

[Ene16] Enerxis. *Nominations & allocations*, 2016 (accessed June 13th, 2016).

[Gas16] FNB Gas. *Market Areas*, 2016 (accessed June 13th, 2016).

[GW97] Jean-Michel Guldmann and Fahui Wang. Optimizing the natural gas supply mix of local distribution utilities. *European Journal of Operations Research*, 1997.

[HL01] Frederick S. Hillier and Gerald J. Liberman. *Introduction to Operations Research*. McGraw-Hill, New York, 7th edition, 2001.

[Hol08] Alan Holland. Optimization of Injection/Withdrawal Schedules for Nautral Gas Storage Facilities. *Applications and Innovations in Intelligent Systems XV*, 2008.

[HWW11] William E Hart, Jean-Paul Watson, and David L Woodruff. Pyomo: Modeling and Solving Mathematical Programs in Python. *Mathematical Programming Computation*, 3(3):219–260, 2011.

[Inc12] Free Software Foundation Inc. GNU Linear Programming Kit, 2012.

[Kar84] Narendra Karmarkar. *A new polynomial-time Algorithm for Linear Programming*, 1984.

[Mar52] Harry Markowitz. Portfolio Selection. *The Journal of Finance*, Volume 7, 1952.

[Mee13] Mark M. Meerschaert. *Mathematical Modelling*. Elsevier, Waltham, 4th edition, 2013.

[Mid07] Kjetil Trovik Midthun. *Optimization model for liberalized natural gas markets*. PhD thesis, Norwegian University of Science and Technology, 2007.

[MMNSD98] Mina R. Marinovic, Dragana D. Makajic-Nikolic, Milan J. Stanojevic, and Lena S. Dordevic. Optimization of electricity trading using linear programming. Technical report, Faculty of organizational sciences, University of Belgrade, 1998.

[Sai16] Peter Sainsbury. *Natural Gas: Top 10 Most Important Price Drivers*, December 2015 (accessed June 13th, 2016).

[TRFM07] Asgeir Tomasgard, Frode Rømo, Marte Fodstad, and Kjetil Midthun. Optimization Models for the Natural Gas Value Chain. Technical report, Norwegian University of Science and Technology, 2007.

[Wil13] H.P. Williams. *Model Building in Mathematical Programming*. Wiley, 2013.

[ZS09] Ines Zenke and Ralf Schäfer. *Energiehandel in Europa: Öl, Gas, Strom, Derivate, Zertifikate*. C.H. Beck, München, 2nd edition, 2009.